SmartRemote: An implementation and survey of a gesture-based media remote controller

Nicholas Cumplido 826488
Supervised by Dr. Jennifer Pearson and Dr. Simon Robinson

Abstract

IOT devices are quickly becoming familiar to everyone, with most people owning one or more in their home. Smartphones being even more common among tech users, these devices hosting numerous sensors that read various types of analogue data. It isn't common for users to have their home IOT devices linked to their smartphones, the use of gestures as part of mobile-spatial interaction is also uncommon. We've explored literature surrounding this area and conducted our own user studies before developing our own prototypes. Our main prototype is for use with a Google Chromecast, the features in this prototype were based from findings in our initial study and findings from the literature. Further user studies were performed with the prototypes developed, the data from those studies were evaluated against the findings of the initial studies.

Project Dissertation submitted to Swansea University
in Partial Fulfilment for the Degree of Masters of Science

Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

Declaration

This work has not previously been accepted in substance for any degree and is not being currently submitted for any degree.

December 5, 2020

Signed:

Statement 1

This dissertation is being submitted in partial fulfilment of the requirements for the degree of a MSc in Computer Science.

December 5, 2020

Signed:

Statement 2

This dissertation is the result of my own independent work/investigation, except where otherwise stated. Other sources are specifically acknowledged by clear cross referencing to author, work, and pages using the bibliography/references. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure of this dissertation and the degree examination as a whole.

December 5, 2020

Signed:

Statement 3

I hereby give consent for my dissertation to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

December 5, 2020

Signed:

1 Acknowledgements

I am incredibly grateful to have had such amazing people around me during this project and my academic career, without them I would not have been able to complete, or even start this dissertation.

Firstly, my supervisors Simon and Jen, who have been my mentors and supervisors since my undergraduate studies. I will always be thankful for their patience, time, effort and support that they have given me throughout the past few years. They have had an immense impact and they will remain to be my role models.

My family, who have always made me a priority and have helped me immensely throughout this project, thank you for your support and patience.

Without these people I would not have achieved what I have so far, thank you.

Contents

2 Introduction

Smartphones are often underutilised by many of its users and applications. Although those devices are capable of performing many simple or complex tasks such as running a wide range of apps and games, online banking, web browsing, video calls and word processing, more can be achieved by utilising the sensors that are embedded into our devices. Our smartphones are home to a rich variety of hardware and components, including silicone processor chips, LED lights, AMOLED screens and antennas. Despite its current uses, such as the gyroscope prompting the screen's content to rotate, the light sensor interacting with the screen to adjust the brightness according to light conditions, more can gained from accessing the sensors in our devices. Hardware can further enhance the users' experience and provide more functionality by connecting the physical to the digital space, primarily through the use of the embedded senors. By using our smartphones for mobile-spatial interaction, users enter a hands-on, head-up type of interaction which offers a whole new experience for users.

2.1 Motivation

It is clear that gestures are a natural modality for some tasks[1], this has been explored in some literature. Some academic and commercial implementations focus on hand/motion detection, rather than using a graspable interface such as a smartphone. An example of this form of gesture interaction is Microsofts Kinect, its focus is on video games which is an area that offers opportunity for further research in more than just gesture recognition. Some researchers claim that user interfaces among smart home devices remain complicated, and those devices become smarter and more interconnected in home environments[10]. In an aim to provide a solution, research has paired those two concepts, gestures being employed to reduce the complexity of device interfaces. Such research includes Ambient Wall[10], a unified interface that allows user to view and control devices through hand gestures. Smartphone devices host a rich set of sensors which together, can form a graspable interface for users to make gestures and control smarthome devices.

2.2 Aims And Objectives

In this document, we conduct our own research into exploring the benefits and drawbacks of using gestures to control smart home devices. Our aims are to further understand and contribute to elements found in the literature, this includes: feedback, the effects of different types of modalities, gesture association and retention, which gestures do users associate with which command and how many gestures and associations do users remember? Firstly, we must understand current gesture usage and smarthome trends. We aim to do this by creating and distributing a questionnaire that asks about these two subjects. Another aim is to develop a prototype using data from study one, this prototype can then be used in study two. Our final aim is to conduct user studies with the prototype, we can evaluate the findings form this study against the findings in the relevant literature.

2.3 Summary

Multiple prototypes were created, one prototype involved scanning a network for IOT devices, another prototype exclusively interacted with a Google Chromecast. The setting in which the application would be used in was considered. The setting was assumed to be a living area of a house, where there is one digital display and multiple seats surrounding and facing it. A feature was implemented that allows the user to set a bearing for the digital display in relation to the user and the device. Both prototypes can recognise basic gestures such as tilting and flicking.

This is done by using the accelerometer and gyroscope sensors. The software that has been developed for recognising gestures uses hard-coded values as part of a threshold implementation. The Chromecast implementation allows the user to cast a video from a selection, play and pause that video, change the volume of that video, all through performing gestures.

3 Background

As with most aspects of technology, we see rapid changes and growth among the hardware and software in computing devices. This section explores the academic literature and commercial products surrounding gesture-based devices. The themes of this section build the foundation for our implementation and studies. We explore the hardware, algorithms and user data created and produced by researchers in this area.

3.1 Gestural Interaction

Smart home devices are increasingly ubiquitous and their interfaces new and varied. Following ubiquity, many types of devices are interconnected, communicating and interacting with each other to create a smarthome. The user requirements in smarthome settings gathered by Kim et al.[10] in 2011, suggest interaction issues were present with smarthomes and the devices that form smarthomes. The authors found that users often waste time searching for the remotes of their devices and those participants of the study found those remotes to be so similar that they confuse their users. Users face similar issues when faced with the outputs of such devices, users often found themselves redoing tasks because they did not recognise the alarms and alerts from the devices and their interfaces.

Regarding the devices, the study meant that the authors were able to state that "The main issue is how to interact with them".

It was clear to the authors that to solve the issues found in the studies, they needed to unify the different controllers and outputs into one recognisable and intuitive interface.

One might think that a solution to this would be to use a smartphone device. This solution which can be inexpensive, it is even more ubiquitous than smart home devices and increasingly intuitive also.

The authors however, decided to implement and trial three systems for smarthome interaction. They decided to investigate three different user interfaces, a PC, a media terminal and a mobile phone.

They also considered other implementations which were very similar to their own research.

Firstly, GeeAir [17], a universal multi-modal remote controller for smart home appliances. This approach used various input types. These were: gesture, joystick, button and light.

They considered Cristal[22], a collaborative home media and device controller on a multi-touch display. The authors state that to control consumer electronics, a user is forced to switch between numerous different controllers, many of which support single user interaction. By unifying the various remote controls into a collaborative integrated remote control, the authors developed CRISTAL (Control of Remotely Interfaced Systems using Touch-based Actions in Living spaces). This implementation came in the form of a gesture-based interactive tabletop, allowing users to to control their multimedia devices through a virtually augmented video image of the surrounding environment.

The link between CRISTAL and Ambient Wall is clear, the status of the 'Smart space' is visible through a digital output and the input being gesture-based and through the same output. Although CRISTAL uses a touch screen approach which is familiar to a large number of people, Ambient wall uses a more natural modality in the form of spatial gestures.

This implementation is almost natural in everyday settings as it takes the form of a tabletop. An obvious issue is blocking the view of the screen by users resting everyday objects such as food or drinks on top of it, sometimes maybe damaging it. The authors recognise this issue by suggesting that this type of implementation be reproduced on a smartphone. Although the authors claim that smartphones are not collaborative, this alternative implementation is further supported by practical aspects. CRISTAL requires users to be situated in positions that are in close proximity of the device, this means that the device must also be situated in a practical position to its users. This aspect of interaction encourages social interaction as users are brought to the device in order to interact with it. There is scope to explore the collaborative possibilities and practical parts of this type of interaction through the use of smartphones, as indicated by their suggestions for future work. There are significant differences between visual feedback on a mobile device compared to a large screen, such as the one used in CRISTAL.

Upon a task analysis of these publications, the authors of Ambient Wall concluded that "users feel uncomfortable and unfamiliar by using these physical interfaces". Although being a physical device, smart phones could be a suitable solution that users might prefer. Ambient Wall brings the interfaces of the different devices together through the projection of context-aware information onto a blank wall, this allows users to monitor what they want to see about the connected devices. It also allows the user to control devices such as a TV.

The overall system comprises of two main components, the display and the gesture recognition. The display uses a projection device to project the context-aware information onto a wall, this information represents the current state of the smarthome. By using this display, users can monitor their home by viewing components of their smarthome when they are needed. The authors decided that the best place to display this information is a ceiling or a wall "because users spend most of their time on the couch at home". The digital display would show information and the status of the IOT devices in the users home. The authors even used APIs from social media networks so the users social feed could be displayed through the display. Examples of statuses displayed include, the remaining time on a washing machine, temperature of the home and any alerts or alarms on certain devices.

The data from their studies showed that users "Spend valuable time searching for controllers". The authors' solution for this was to implement a single controller/input method based on hand gesture recognition. The researchers considered the issue of distinguishing common gestures to gestures required for using Ambient Wall. They were able to solve this as users don't usually point at walls, ceilings or blank surfaces.

They recognised that using multiple devices was "troublesome work", as proven in the results of their user studies. Their answer to part of this inconvenience was implementing the ability for users to turn off all devices at once.

This publication did not give any indication of user studies with the prototype. The only hand gesture demonstrated with this system was an illustration of a user making a circular motion with their index finger. It is clear that the prototype developed solved issues of the problems with multiple unfamiliar controllers by unifying the inputs and outputs of those controllers into a digital display.

This digital display lacked portability, this results in the user needing to be in the same location of the room each time they wish to use the device.

This implementation could be considered as expensive and installation and calibration complex. Post-development user studies would have revealed thoughts and feelings regarding hand-gesture recognition in this setting, such as the preference of using something tangible to control the devices.

Academic research from 2008 recognised the issue of the unfamiliarity of mobile gestures (Linjama et al.)[12]. By developing an interactive tutorial application based on gesture input, The authors were able to enhance the user's experience and learning of using gesture based

devices. This was achieved by using a physical tangible visual object in a mobile device. The device they developed used the embedded accelerometer as input, the input from the accelerometer consisted of two signals. One signal represented gravity orientation and the other signal gave inertial data generated from fast movements. The aim of the research was to display this data to the user in a meaningful way. They achieved this by displaying a digital 3D element on the mobile device's screen. This 3D element took the form of a cube, this cube rotated in relation to the movements of the device. The authors believe that there is potential for future mobile phones equipped with accelerometers to use a combination of interactive elements. This can be utilised by the device in this project, mainly through vibrations from the handheld device. As the user can see the action being completed through the content on the screen changing, the user could benefit from feeling feedback from the device too. This is something I believe would be worthwhile to implement and study.

3.2 Remote Interaction

Research conducted by Robinson et al.[19] investigated how participants could blog locations using gestures, their findings are documented in their paper Point-to-GeoBlog. For a participant to 'geoBlog', they would point a device to a location and perform a gesture. The author describes the gesture as "casting out a net and drawing it back in to the correct position". The authors used SHAKE(Sensing Hardware Accessory for kinaesthetic Expression) which is a sensor pack for real-time recording of sensor data. The wireless device contains a compass, three-axis accelerometer, magnetometers and angular rate sensors.

Two prototypes were developed in this research, the first prototype focused on gestures and visual feedback of a map. The other being a lightweight version, which focused on gestures only. To mark geographical points, participants would point the device to an area of interest and perform a gesture. The visual prototype used an arrow which was overlaid onto a aerial photo of the users current location, this would assist the user in knowing where they were facing. Marking the area of interest would require the user to tilt the device towards or away from the user, the user would press a button to save that location. To mark/save a location, the user would tilt the device. The angle in which the user tilted the device determined the distance from the user the point of interest was. The tilting gestures that were used in their research would be worth implementing and investigating in the setting and context of my application. The level of exertion is similar to our work, no rapid or strenuous movements are required for the gestures. This gives us reasoning for implementing similar gestures. The user could also record locations without a line of sight to the location. This would be useful in our prototypes as the user could change the state of the media content before they entered that room. However, this raises the issue of providing feedback to the user, such as which modality should be used and for which interaction, if any feedback should be given at all.

The device in this paper was awake and powered on at all times, this is worth considering when evaluating the content on the screen of the device. However, it was noted by the authors that there was a "potential added burden of visual feedback". This is loosely related to the issue of listening for gestures. Our implementation will need to know when to expect gestures so it doesn't accidentally recognise a command. For example, if a user were to accidentally knock/drop their device or when they want to use it for a different purpose. An obvious approach to solving this would be to implement buttons, implementing 'listening' and/or 'save/finish' buttons are feasible, however this would interrupt and impede interaction. The buttons would need to be visible on the devices screen, relating to the conflicting evidence of visual feedback from other research that is included in this document, and results from the survey that I have carried out. For some interactions with my applications, the visual feedback could be the result of the device the user has just interacted with. For example, the user would

not need visual feedback from the device if the user changed the channel on the TV, or if they changed the volume with the app. Visual feedback from the application would be useful if the user is performing an action that does create any noticeable feedback, such as turning the light off in the next room.

A standard back-propagation neural net was used to recognise the gestures. This method of recognising gestures looks for stability patterns in the sensor data. For example, the user taking the device out of their pocket and then holding it steady to begin the gesture. Once recognised, the compass reading and GPS readings are combined.

To evaluate the prototypes, the authors made use of Nasa's task load index (TLX). This form of user study composes of 6 Likert scales, each scale is used to rate a different type of exertion that can be applied to the task of the study. The types of exertion include physical, mental, temporal, performance, effort and frustration(See figure 12).

Points to takeaway from this study are: visual feedback, gesture types and the use of Nasas TLX.

3.3 Technical Background

In 2013, researchers published[1] a paper of a gesture recognition platform. In this publication, they implemented a dynamic time warping algorithm to create a mobile instrumented platform that interacts with smart objects using gestures. This implementation allowed users to assign their own gestures through a training algorithm. The algorithm also recognises gestures made in real-time. The only hardware components mentioned are accelerometers from a smartphone device, the data from the accelerometers are used to recognise gestures.

By pressing the name of the smart object on the screen of the device, a user can learn information about that device, such as gesture configurations. The user can also control the device by pressing its corresponding button.

They based their justification of implementing a feature that allows users to assign their own custom made gesture off of Kela et al's[9] findings of a gesture-based study. Kela et al's questionnaire-based study of 37 participants revealed that different people used different gestures for the same task. This finding reinforces the requirement for users to be able to assign their own gestures, this then suggests that implementing a time warping algorithm for gesture recognition is a suitable solution.

The authors based the dynamic time warping implementation on methods from Myers et al's[14] work on the performance trade-offs in in dynamic time warping algorithms for isolated word recognition. The basis of the algorithm is to solve optimisation problems, it minimises the time-normalised distance between two pattern series. The minimised distance is the similarity between the gesture template and the data from the device's accelerometer.

By using research performed by Ko et al.[11], the authors were able to determine the minimum selection for using template selection. The recognition system creates templates during the training stage by using the intra-class dynamic time warping distance, which is the sum of DTW distances between one certain sample and all the other samples within the same class. The sample with the minimum intra-class DTW distance is selected as the template of this class.

One of their implementations involved using an Android smart phone, they measured that this implementation had an accuracy rate of over %98. This number was produced by testing the implementation with 21 different gestures that were performed 5 times for each gesture. This was completed by 11 different participants. These gestures involved directional movements, letters and signatures. They tested three types of distance measurement algorithms: Euclidean, Chebyshev and Cosine. The Euclidean algorithm proved to be the most accurate, this algorithm allowed instantaneous recognition of gestures with no noticeable delay to the user. An Arduino

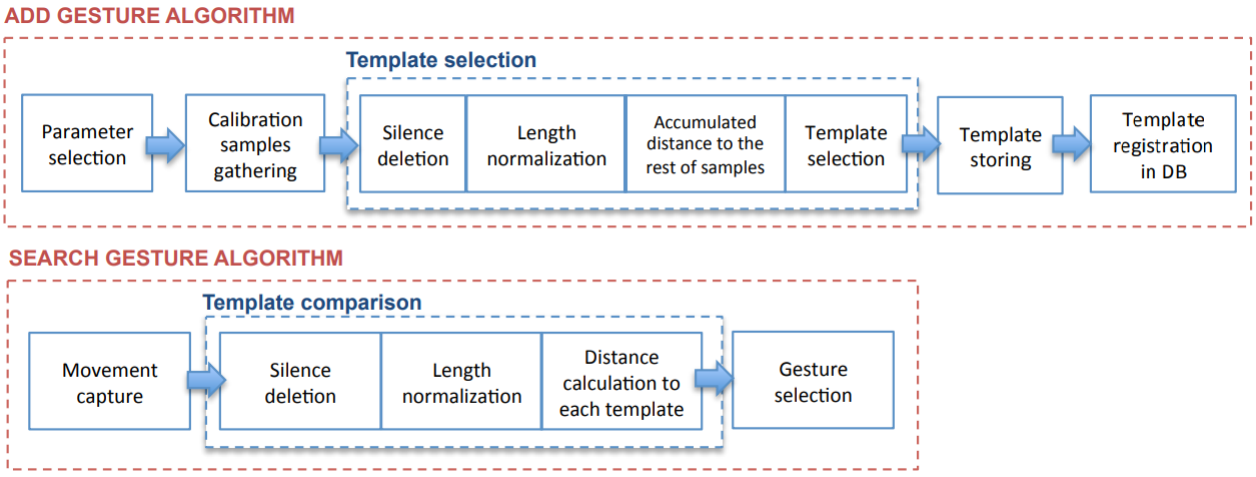


Figure 1: Stages of the dynamic time warping algorithm, taken from Barnardos et als' gesture platform paper.

was used to "smartize" two smart objects, a blind and an LED light. Once connected to a server, the user can retrieve the modules of the smart object. Each module contains contains configuration data: an IP address, the port number to connect the object, perform-able actions and associated execution messages. Although the Arduino was connected to a relay, and while it may be worthwhile exploring and recreating this implementation, using a basic router instead of an Arduino may be better as the software in off-the-shelf routers may be suitable enough. Android offer various APIs that manage WiFi and peer to peer connectivity therefore, reducing the need for developing custom router software. For a user to add a new device to the 'smart space', they are required to add the configuration files in the server and implement the logic for the object. A user must assign and train a gesture to a smart object in order to use that object, this will be prompted if the user attempts to perform a gesture on a device that has no assigned gesture. For a user to train a gesture, they must use an external application which contains the algorithm for adding a gesture, this algorithm then produces a gesture template which is then stored. Templates can be generated by performing a gesture, but the quality of the template improves when the gesture has been performed at least three times or more. Users can access the list of objects once the gestures have been assigned to the objects and control the target object. Acceleration data is retrieved while the user holds their finger on the screen, the device then enters "gesture-control-mat mode" and the user can then control objects through gestures until they release their finger.

A study performed by Robinson et al.[20] investigated the use of a mobile device to physically interact with paper posters, the study relied on users approaching the and tapping the poster with a mobile device. The publication considered various methods of connecting to, or recognising the display, such as NFC and RFID tags (Rukzio et al. [21]). The user would interact with the poster using multiple techniques, sensors in the users smartphone would be used to detect neodymium magnets(Bianchi and Oakley [2]) after scanning the QR code to uniquely identify the poster. The user is then prompted to calibrate their phone with the poster, this is so the information is stored for future interaction with that poster. Uses for this system include a self-help health information poster. This poster allows the user to tap symptoms that are displayed on the poster, after tapping the poster the user would receive audio information about that symptom. If the user wants information about a symptom from an expert, they would record a message by speaking into their phone and by tapping the poster, the recording would be sent. This publication investigated methods that a mobile device could use to connect to physical objects, particularly around the use of magnets and QR codes. While these techniques and methods are possible for my proposed project, they would require the user to

approach the digital display for each interaction and command they would like to perform. It would be more convenient for the user to operate the digital display with its original hardware (such as a TV remote) than to approach the display. Although it was unfamiliar among the general population in 2015(Ozkaya et al.[16]), since its invention in the mid 90s(Robertson et al. [18]) the use of QR codes could be more convenient in a larger setting. For example, a user could scan a QR code on a display in a public setting, this would aid in identifying which display to connect to(implemented by Robinson et al. [20]. This feature could then extend to home use, QR codes could be used to identify different displays that are around the users home, perhaps even IOT devices. If QR codes were implemented in my project, it would be beneficial to prompt the user on the use of the codes within the application. In household settings. It may be worth implementing a dedicated home network, similar to the *FEMINITY* network created by(Ouchi et al.)[15].

Recognising gestures require multiple hardware components and methods, as proven by Cho et al.[3]. Some physical components used in this publication are available in smartphones, or at least produce the same output, such as:

- Gyroscopes, one for each axis (X,Y and Z).
- Accelerometers (also one for each axis - X, Y and Z).
- An analogue to digital converter, this is to convert physical data (eg. movement) into digital data (ones and zeros) that the device can perform calculations with.

The paper stated that "Accelerometers measure accelerations and gyroscopes measure angular velocity.", these are the values required to recognise physical movements. As these two sensors are present in smartphones and the application programming interfaces to use the sensors are publicly available, it is possible for us to develop a smartphone application that uses these sensors.

The authors made use of a serial port interface to "transmit sensor data to a PC when collecting data and training gesture recognizers.". Practical alternatives to serial port interfaces exist to recognise gestures, such as a user 'recording' or calibrating their own gestures on the application. This would then connect to the display wirelessly.

The authors employed a gesture recognition algorithm based on Bayesian networks[23], this mathematical model proved to yield a high accuracy at recognising gestures, as high as 99.2%. Bayesian networks allow reasoning under uncertainty[8] by using a directed acyclic graph. The graph is composed of states that represent variables and directed edges between them. More directly, Bayesian networks are probabilistic graphical models. They are useful for predicting the cause of an event from a set of possible causes. It was applied to gesture recognition in the publication above by using a set of gestures that the user could make as the set of possible causes. The event would be the readings from the sensors in the device. The computational complexity of a Bayesian network is *NP-Hard*(Non-deterministic polynomial time)[5]. As the device was custom and purpose-built, it could accommodate the computational model. Taking the computational complexity of Bayesian networks into consideration, along with implementing the model in a device that runs multiple processes for various purposes, it seems impractical and unnecessary to use Bayesian networks to recognise gestures.

Freeman et al.[7] conducted research in 1995 that explored how an individual could control a television set by hand gestures. This relates to this project however, the focus of the authors was to tackle two issues: recognition of hand gestures by the computer, and the memorisation of gestures by users. The authors implementations were able to encounter the issues they set out to solve. Technology has progressed a significant amount since the publication of this paper, this allows us to utilise a far wider range of hardware and software to develop a solution. The authors learned that the participants found certain gestures tiring after extended viewing,

this was because users had to hold their hand up for the recognition system. The authors suggested a trigger command for further work, this is something that could be tested in our implementation.

The Logitech spotlight advanced presentation remote is more than just a standard presentation tool, this device uses gestures so the user can: move an on-screen cursor, control a spotlight, highlight content and change the volume and control the video content on the screen. The main components used to complete the actions are the devices accelerometer and gyroscope. This commercial product is built upon and uses the work mentioned earlier by Cho et al.[3].

The Nintendo Wii mainly uses gestures as its input, with secondary input coming from buttons. These two forms of input are utilised by a remote control that the player waves and shakes while they play a game.

Players can hold the remote as if it was a racket or a bat, or as if they were wielding a sword. This meant that players would use a wide range of movements to play the many different games that were available. Players would swing, flick, shake and rotate the remote.

Microsoft's Xbox Kinect is a marker-less based implementation of gesture recognition(Zhang [24]). It utilises various sensors such as:

- A four-microphone array which provides full-body 3D motion capture, it also gives facial and audio recognition.
- Depth sensor
- Colour camera

The author claims that "The Kinect sensor offers an unlimited number of opportunities for old and new applications" and that additional research areas involving the Kinect involve hand-gesture recognition. These two points create strong reasoning for implementing my proposed project with a Kinect system. However, multiple issues arise regarding gesture recognition, such as the various gestures that can be performed and communication between the sensor's portability to connect to different digital displays. Without memorising and using more gestures (an issue discovered by Freeman et al. which was mentioned previously), Using this sensor would limit the capabilities of connecting to other IOT devices. Research conducted by Freeman et al.[7] that has been mentioned earlier, described issues relating to participants becoming tired while using their hand for gestures with sensors. The author suggested that a trigger command be implemented in further work, the Kinect system would be an appropriate system to implement such a feature. A 'wake' gesture requires a deliberate and specific gesture to prevent accidental activation. To allow this feature to work for normal use, it would need to be powered on at all times, waiting for the command. Users may feel uncomfortable or uneasy with a sensor containing cameras pointing at them for long periods of time, similar to how some users feel about the Amazon Echo and its microphone sensor. Another issue described by Freeman et al. was the amount of gestures that could be used, not only would there be many gestures but these gestures would have to be memorised. The Kinect device is designed to connect to an Xbox gaming console, significant modifications would have to be made for it to connect to different displays. Especially if there is one Kinect sensor in a household of multiple displays. These factors limit the availability of wider use by people because of cost, usability and practicality. Unlike the smartphone implementation I plan to complete.

The literature covers various products for different purposes, each with differing implementations based on their contexts and applications. Naturally, implementations are influenced by the technology available at the time of design, this is evident in the work by Freeman et al.[7] which was published in 1995. Although the the Kinect sensor shares the same principle as the

system in the publication, the quality and quantity of the Kinect's sensors are superior. This factor could permit solutions to the issues mentioned in the publication.

3.4 Conclusion

The use of smartphones as a controller based on gesture input provide solutions for the issues mentioned above. Modern day smartphones possess the hardware to recognise gestures without the use of cameras or static external sensors, this allows the controller/remote to be portable. The wireless capabilities of smartphones would permit users to use the digital display at a distance, unlike *PosterPointing*, which was mentioned earlier. An important issue to take into consideration is the unfamiliarity of using gestures with a smartphone among users. This could be solved by allowing users to record and create their own gestures on their device and then assigning their chosen gestures to the commands available. This solution would then influence the issue of users remembering the many commands, partly through active learning and going through each command and performing an action. As proven by Macken & Ginns [13], performing gestures has a positive impact on learning. The content covered in this section provide direction and justification for the methods and practices for my implementation.

4 Study One: Current Gesture Usage

In order for us to understand the users requirements relating to our project, we created and distributed a questionnaire of 16 questions which we received 124 responses. Alongside some of the findings in the literature mentioned previously in this document, the data from this study influenced the design choices that we made. We aimed to get data about participants gesture habits and the reasons for performing those gestures. We felt it necessary to ask users about how frequently they lose their TV remote, although this was briefly mentioned in Ambient Wall, it was appropriate to get specific data for the context of this work.

The questionnaire also asked for the following information:

- Average level of difficulty for understanding the interfaces of your devices and appliances
- Which smart home IOT devices are owned and how many
- Gesture assignments
- Ease of use and accessibility, phone accessories
- Device feedback

4.1 Methodology

The groups that the questionnaire was posted in were community pages on Facebook, and questionnaire/survey exchange forums on Reddit and Facebook.

Sixteen questions were asked in total, there was a mixture of open, closed and scaled questions. Participants were first asked if they make gestures with their smartphone and if so, why.

The only definition of 'gesture' that was given in the first question: "Do you ever make gestures with your mobile device? Eg, shake, flick, rotate, If so how often?".

4.2 Results

Demographics of the participants of the questionnaire were not recorded by the survey however, judging by the number of participants and distribution channels, it can be assumed that the participants are from varying backgrounds with varying levels of technical literacy.

50.8% of the participants don't make any of the gestures defined and only 34.7% of participants make gestures less than five times a day.

The most common reason for making gestures with a mobile device was to rotate the device to view media such as movies and photos. We also received responses where the participants described gestures such as "pinching to zoom".

%47.6 of participants never lose a remote control over the course of a week, %41.9 of participants lose a remote control up to five times a week with the rest of the participants losing a remote control more than 5 times a week.

Contrary to Ambient wall, responses show that the difficulty of understanding the interfaces of household devices and appliances was low. With %73.4 of participants finding it easy to very easy to understand the interfaces.

Amazon's Alexa smart home device was owned by %48.4 of the participants, %36.3 owned a smart TV and %33.1 don't own a smart device. Only one participant owned a Chrome cast.

From the participants that owned smart devices, least one of those devices was connected to a smartphone, %47.6 only having one device connected and %16.7 having more than 3 devices linked to a smartphone.

The responses revealed that people have their smartphone device with them for the majority of the time that they are in their house: %53.2 most of the time and %24.2 all of the time.

Some interesting responses were given to the question: "Would you prefer to use your smartphone as a remote control for your TV and/or smart home devices?". While over a third(%35.5) of the responses wanted to keep remote controllers and smartphones separate, almost two thirds would like a combination of the two or to have the choice.

46 of the 99 participants that would use their smartphone to control the content of their TV would prefer to press buttons than to use gestures. Only 5 would prefer gestures over buttons, 32 participants wanted both options while 16 had no preference.

Participants were then asked what motions they would assign to the following commands:

- Stop
- Fast forward and rewind
- Next/Skip
- Volume up/down

For each command, an average of 97 participants gave an answer.

This section of the questionnaire yielded interesting responses. The answers showed that participants thought that they were being asked about hand gestures, not mobile-spatial gestures. The majority of the answers for each command were similar for example:

- Participants would shake the device or raise a hand to stop the content that was being displayed.
- Rotating and making circular motions was a popular answer for fast forwarding and rewinding content.
- Swiping right and and finger pointing were common answers for skipping content, some answers for this assignment were the same as some of the answers for fast forwarding and rewind however.

- Assignments for changing the volume were very similar to the assignments for fast-forwarding and skipping. However, the actions would be performed vertically, such as raising and lowering the device or pointing upwards/downwards.

Many of the answers given for each assignment were to use buttons, some explicitly said the buttons on a smartphone device as some commands(fast-forward) were already "fiddly enough".

For the final command, some participants could not think of any more gestures to assign.

Some of the assignment responses were widely varied and unique, some assignments required more exertion than others, such as shaking the device, thrusting forward and

Some responses referred to current implementations like Google pixel 2s squeeze sensor.

When asked on how confident they would be remembering the different gestures required to control devices, there was an even spread of responses against five levels of confidence(confident to not confident).

When asked about accessories for their smartphone device, 85 participants use a case, and 12 use a pop socket, 33 did not use any accessories. This question was aimed to give an understanding/insight into the possibility of damaging the smartphone/device being used to perform the gestures.

%61 of participants prefer visual feedback from their device, %41.5 for haptic feedback in the form of vibrations and %28.5 for audio. This question was asked to give us direction in investigating the types of modality that can be used in our implementations.

4.3 Analysis and Discussion

The findings from our questionnaire contradict the findings shown in Ambient Wall[10]. Out of 124 participants for our questionnaire, almost %50 never lose remotes and over %40 lose the remotes for their appliances and devices less than 5 times a week.

Over %70 of participants claim that the level of difficulty in using smarthome devices and household appliances is easy to very easy.

Very few responses mentioned shortcuts linked to gestures, such as shake to undo. This could be due to the possibility of unawareness of gesture-based features.

Many factors need to be considered when comparing the data from Ambient wall to our data. Our data has been gathered 9 years after the publication of Ambient Wall(2011), this could show that users are becoming more literate and familiar with smarthome devices, appliances and maybe interfaces in general.

It seems highly likely that defining the term 'gesture' would have made the responses more reliable, some responses to some of the open answer questions showed that participants misunderstood the theme of the questionnaire. Although the title of the questionnaire was "Remote control gesture questionnaire", some participants associated gestures to hand gestures rather than the intended motion gestures made with a phone or a controller.

After analysing and evaluating the data from study one, we were able to implement features in our prototypes that would allow us to build upon the data gathered and the data from the literature. For example, by knowing which gestures that participants would make for certain actions, we could set a default gesture for each command.

5 Prototype development

This section describes and explains the prototypes we developed and the justifications behind them. We attempt to create a smooth user experience that avoids and solves some of the issues found in literature surrounding this subject. This work also aims to build on some of the work

completed by others, taking some aspects that we feel could be applied to different contexts. Primarily, our choice in developing a smartphone application was influenced by the number of users that kept their smartphone with them when at home, when asked in the questionnaire. Almost a quarter of the participants that took part in our questionnaire said they always have their smartphone with them when in their home and over half of those participants said that they have their device with them most of the time that they are home. This is also justified by the responses to the question: "Do you ever lose the remote control for any of your devices or appliances?". Over %50 of participants lost a remote control at least once a week. Results from Ambient Wall and research into the Kinect sensor proved that hand-recognition to be an expensive solution and one that wouldn't be feasible to setup for its intended use. Although it was not found in literature, we decided to look into Tensorflow lite as a method of gesture recognition. This is because machine learning had not been implemented in previous work. Also, Tensorflow lite is designed for use with mobile devices. Some of the devices in the literature that has been covered had built-in displays. Some of those devices displayed some information, ActionCube displayed a digital cube to show movement and point-to-Geoblog displayed an arrow to represent magnitude.

Similar to point-to-Geoblog, we decided to develop multiple prototypes where the content of their displays are different. As a result of one of the questions asked in the questionnaire, we decided to investigate users preferences regarding the different modalities that can be offered by a smartphone device. The researchers of Actioncube[12] found that users had a positive experience in using a device that had moving content on its display. Alongside user experience with digital content in point-to-Geoblogs[19] implementation, we believe that it would be worthwhile investigating the affects of different modalities on gesture interaction. Relating to the content of the screen, we decided to ask participants of the studies where their attention was focused during interaction with the device. This was asked to discover the possibility of investigating a different aspect of 'second-screening'. The default gestures that are included in our prototypes are based from the findings in our questionnaire.

5.1 Design overview

We started developing the prototype by thinking about where in the home it would be used, starting with the living room. In most homes, the main feature of a living room is the TV, with multiple places to sit surrounding the front of the TV. Although some of us have our regular spot to sit in the living, we may move around or sit somewhere different for a change. We also sometimes pass the remote controller to someone else in the room. These factors must be considered when implementing the application, especially when it might be used with multiple devices. For a user to issue a command with a gesture, we decided that they would perform that gesture in the direction of the device they want to control. We decided against the user being able to select a device using the screen because we want to maintain a gesture-based experience, as well as a heads-up experience. This results in the bearing of the smarthome devices relative to the application, being recorded by the application and accessed when gestures are made. To implement this feature, we chose to use the x-axis value from the game rotation vector software sensor, which gets the rotation vector of the device at the x, y and z axes. We chose this sensor as it is more accurate and reliable that the deprecated orientation software sensor[1]. To save the bearing of a device, the user would swipe to the left on their smartphones screen, point to the smart device and tap the 'Set bearing location' button on the navigation drawer. Although this requires interaction with the screen on the device, we felt that it was suitable to implement this as it is a low-cost and infrequent action that the user would not have to use often.

We also implemented a feature where users can set their own location profiles. This feature

[1]09/2020 https://developer.android.com/guide/topics/sensors/sensors_overview

involves the user creating a profile with a name, for example 'Armchair' or 'Bean bag'. The user would then add a device from a list of previously connected and setup devices to the profile. As the user selects a device from the list, they would have to point to the device and save the bearing, similar to what is shown in the above image. This calibration step should happen infrequently, maybe only once. While this is still a screen-based interaction, it is low-effort and infrequent enough to allow it in the prototype.

A TV remote does not need to be 'woken up' or unlocked like a smartphone therefore, we implemented a feature where the smartphones screen is always awake and 'listening' for gestures while the application is open. This also meets our intention of implementing a low-effort and 'heads-up' prototype.

A priority in this application is to replace buttons and taps with gestures, through this we have the potential to learn how different methods of listening for gestures impact user experience. Previous publications have displayed different ways of listening for gestures, such as handheld devices, motion detection for hand gestures and tapping the screen of a device before and after performing the gesture. Our gesture listening feature is inspired by Robinson et als' Point-to-geoBlog method of completing tasks through gestures. In their work, the user must hold the device horizontally before they can can complete an action. The angle of the device to the horizontal plane determines a value which the user can use for certain actions. By tilting the device horizontally, the device then 'listens' for a gesture where tilting left or right in this horizontal position performs different actions. The gestures mentioned so far are deliberate actions that are not fast or erratic movements therefore, and to avoid recognising a gesture that was not intended to be performed, a gesture listening feature should be implemented. However, flicking is a quick and intentional motion which is uncommon for smartphone users, especially in the context of our work and the context of Point-to-Geoblog. Therefore, it does not seem necessary to implement a gesture listening mode for this gesture, so in our implementation we allowed the flick motion to always be listened for. In the instance of casting to the Chrome cast device, the user must tilt the device forward and in the direction of the device. Similar to flicking, this motion is also deliberate, so it doesn't seem necessary to implement an on/off feature for gesture detection. The application uses sqlite3 to manage and store the database of the devices and location profiles. sqlite3 is a relational database management system that is entirely located in the program. Meaning that it is not located on a server that communicates to a client. The database consists of three tables; a table to store the devices; a table to store location profiles; and a pivot table to link the two other tables. A many-to-many relationship is established with these three tables, as shown in figure 2.

Location profile table

ID	Name
1	Sofa
2	Bed
3	Bean bag

Pivot table

ID	Location_ID	Device_ID	Bearing
1	1	2	
2	1	3	
3	2	1	
4	2	2	
5	2	3	

Device table

ID	Name
1	Bulb
2	Tv
3	Blinds

Figure 2: The many to many relational database between the devices and location profiles tables.

5.2 Gestures

There are a vast amount of gestures that we could make, the limit is the amount of gestures that the human body is capable of. However, only a few are required to match the number of possible controls that can be done with a media player and smart devices.

The default gestures that we included were inspired by the results from our questionnaire and work by Costante et al[6]. Rotation, moving to the left/right, flicking and swiping being some of the most common gestures mentioned in the questionnaire. They made use of 20 gestures, some being similar or the inverse to other gestures as shown in figure 3. The dataset[2] that was generated from their study came from 8 participants performing all 20 gestures, 20 times. The gestures were made with a Sony smart watch and the data generated from the embedded accelerometer. Users would tap the screen of the device before making the gesture, this is so the device would start recording the data from the accelerometer. Timestamps were recorded and the value of acceleration in the x, y and z axes at the time of the timestamp were recorded. Once the gesture had been made, the user would then tap the screen to to stop recording the gesture. Figure 3 visualises some gestures that participant one made.

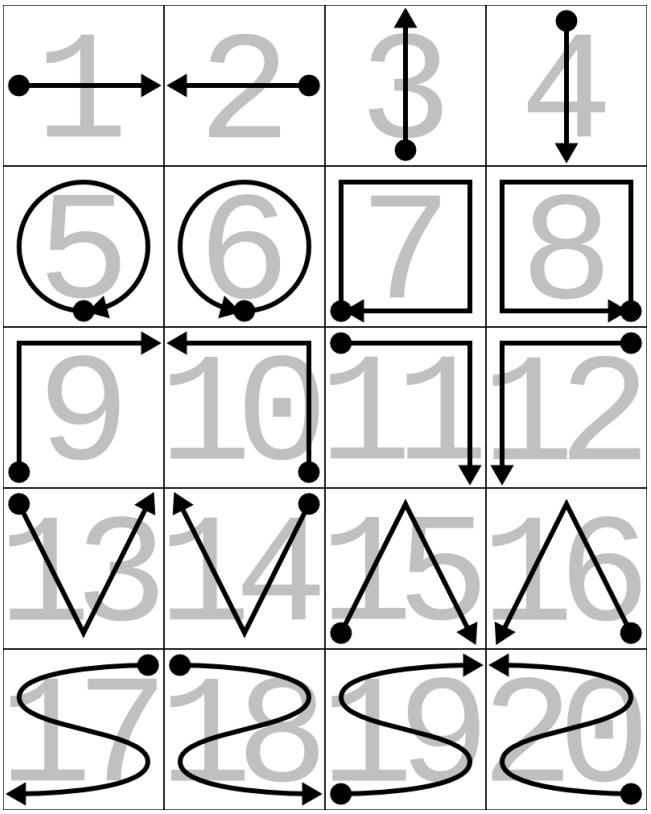


Figure 3: The vectors from Costante's et als' work in which the gestures in he dataset are made.

We did not include such a wide vocabulary of gestures but only used 4 simple gestures(shown in figure 3).

The choice in gestures are justified by the responses in the questionnaire (with rotation being popular) and gestures used in literature[6][19].

[2]https://tev.fbk.eu/technologies/smartwatch-gestures-dataset

(a) Flick up

(b) Tilt down

(c) Tilt and rotate left

(d) Tilt and rotate right

Figure 4: Motions used in our prototype application

(a) Flick (b) Start

(c) Tilt up (d) Tilt and rotate right

Figure 5: Examples of gestures being used with a Chromecast

5.3 Sensors

We make use of the sensors embedded in a Samsung Galaxy S8, the rich variety of hardware available in the device can be accessed by applications through the Android sensor framework and the hardware access layer(HAL)[3]. The hardware accessible by this framework include: accelerometer, ambient temperature, gyroscope, light, linear acceleration and acceleration due to gravity, magnetic field sensor, orientation sensor, pressure, proximity, relative humidity, rotation vector and temperature. All of the previously mention sensors are available in Android 4.0(API Level 14). The previously mentioned sensors can be categorised into three types: motion, position and environment. We also utilise the device's vibrator hardware to give tactile feedback when certain conditions are met. For example, when the volume increases or decreases.

All sensors can be split into two categories, hardware-based and software-based. Hardware-based sensors are the physical components that are embedded into devices, these components take the analogue input that would be changed into digital output. Soft sensors mimic hardware-based sensors, they combine sensor data from multiple sensors to give data that cannot be represented from a single data, such as linear acceleration and gravity. The sensors that we used are the accelerometer, gyroscopes and rotational vector sensors. These sensors measure acceleration forces and rotational forces along three axes, as shown in figure 6.

We focused on using the raw sensor data provided by the sensors. By using the data from the hardware access layer, we can specify the refresh rate through the code. The refresh rate of the sensor data being read impacts performance and accuracy of gesture recognition.

This framework allows us to perform a variety of operations and tasks through the use of classes and interfaces provided by the framework. We used the framework to: create an instance of the sensors required by the application once it was determined that the device had the required sensors, register and de-register listeners for the sensors, and to acquire the raw sensor data.

Recognising some gestures relied entirely on the linear acceleration soft sensor. This excluded the force of gravity and measured the force of acceleration on the three different axis, as shown in figure 6.

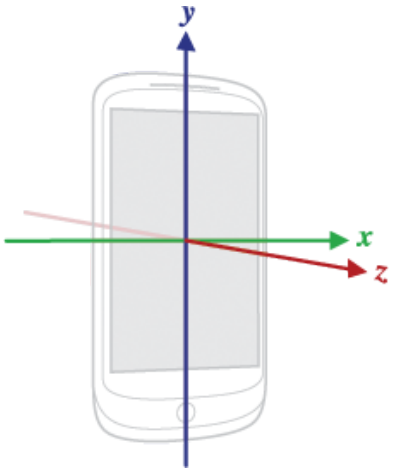


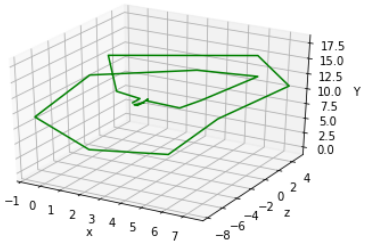
Figure 6: Coordinate system (relative to a device) that's used by the Sensor API, taken from the Android developers site.

[3]https://developer.android.com/guide/topics/sensors/sensors_overview

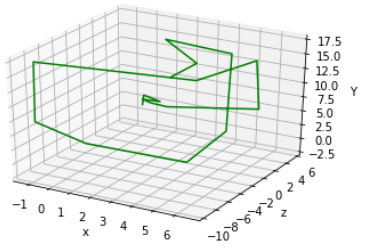
5.4 Gesture recognition

Earlier in this document, methods of recognising gestures from sensor data have been explored in literature, none of which used Tensorflow or Tensorflow lite. Tensorflow is a platform which hosts tools and libraries which developers can use to create machine learning models. These models are created in python using Keras. Tensorflow lite is a more portable version of standard Tensorflow, designed for use on mobile devices. Tensorflow lite models are written as Jupyter notebooks in Python, this then runs in Google Colaboratory. By using the Python library numpy, large mathematical operations can be performed, this can then be used with the Python library matplotlib to plot graphs.

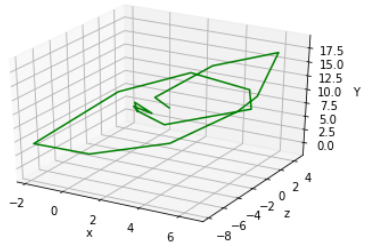
Figure 5 shows visualisations of the dataset mention above[6]. Each point in each of the 3D scatter plot graphs represent the velocity in the x,y and z axis at a certain timestamp. The lines in the graph are connected to two consecutive timestamps, the lines and the plots then visualise the gesture through the data gathered from the sensors. Most of the gestures that were performed could be easily identified by looking at the visualisations, as shown in figure 5.



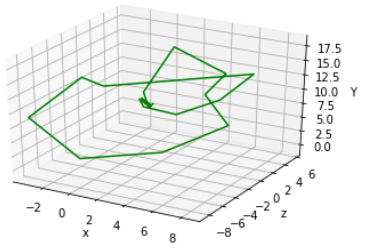
(a) 5: A circular motion



(b) 9: A right angle



(c) 14: A 'V' shape



(d) 17: An 'S' shape going downwards

Figure 7: Gesture data plotted on a 3D graph, by using the gesture grid in figure one, the number in each caption matches a number on the gesture table in figure.

The work presented above was carried out with the aim to utilise Tensorflow lite to recognise gestures made from the sensor data from the device.

The default set gestures we chose to include in the application

5.5 Connectivity

Early implementations of the application were developed with the aim of scanning a network for IOT devices. The application would use APIs such as the wifi manager to get the address of the router it was connected to and ping all of the possible IP addresses. If it got a response, it would save that IP address so it could get information of that device later. Due to the computational power required by the task, multiple threads would need to be created.

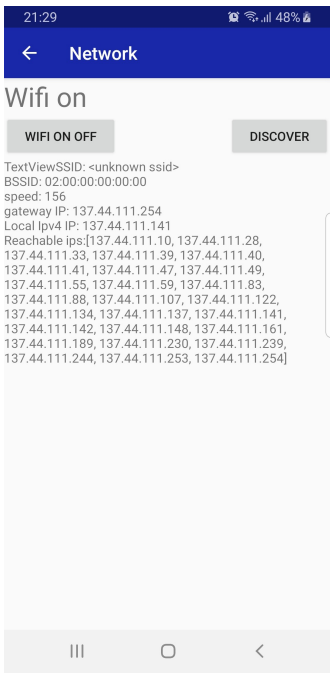


Figure 8: A screenshot of the results of scanning a WiFi network with the application. The reachable IPs are the devices that are connected to the WiFi router.

As development progressed, it became clear that implementing a network scanner in the main application was unfeasible. An alternative solution was to develop an application that is dedicated to interacting with a Chromecast[4]. A separate application was created for use with just a Google Chromecast, it was then merged with the original app. As a result of merging the two applications, we are able to use the sensors to save the bearing of the Chrome cast device and use gestures to interact with the media player.

The below figure visualises the layout of the environment/setting in which the user may use the application and the Cast device.

6 Study Two: User Studies

This section shows the methodology and results of the user studies that we carried out with the prototype. The data generated in this study allows us to measure our implementation against the implementation and data from previous studies.

Under normal circumstances, we would aim to get 30 participants to take part in user studies. However, two participants were able to safely take part.

6.1 Observations

The following studies were performed in-person with two participants, both with a similar level of technical experience and literacy. The average age of the participants was 55 years old, both working professionals in a non-technological field. Their experience of technology and digital applications are limited to office applications such as word processors and emails, as well as some basic mobile applications.

[4]https://developers.google.com/cast/docs/android_sender/integrate

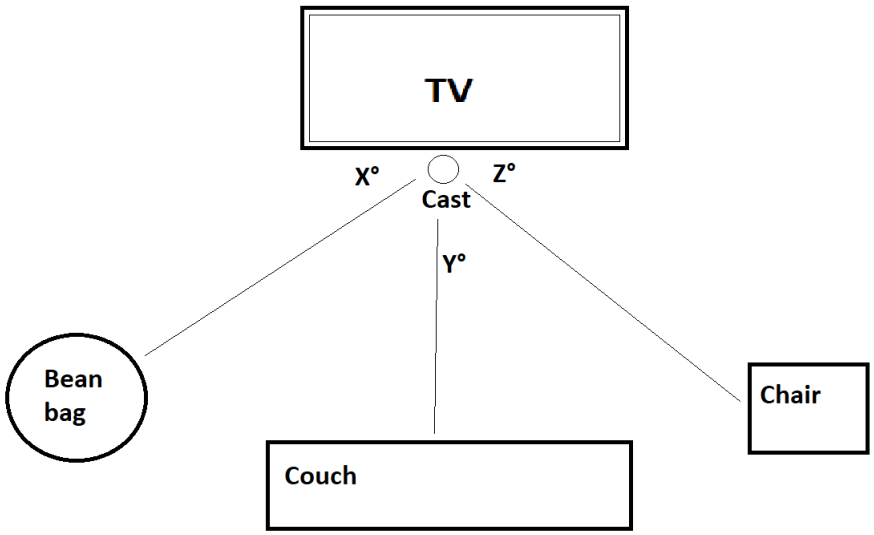


Figure 9: The possible layout of a space where the devices and application may be used.

6.1.1 Methodology

The setting in which the tasks were performed replicated the intended setting of the application: a location that is familiar to the participants(the participants home) and where they can sit and watch TV.

The components of the system was already setup, a Chromecast was connected to a monitor and connected to the local WiFi. The monitor sat on top of a desk and the Chromecast rested on the side of the desk, a chair was placed in front of the desk in a position that the participants could sit comfortably. The equipment was powered on with the Chromecast in ambient mode, the system was waiting for input. During observations, I stood to the side of the participant as I instructed them on the actions they should perform: locating the chrome cast, selecting a video, starting and stopping the video, raising and lowering the volume, then disconnecting cast and shutting down the application.

The application and devices were described, as well as how to complete each action. For example, how to locate the device, cast to the device and then how to control the media through gestures. Each participant was given the same description of the application and devices, this was to ensure that they both had the same level of knowledge before completing the tasks.

Before each observation took place, the apps data was cleared, restoring it to its default settings. This is so each participant could experience the application as if they were the only user, and also to provide unbiased data. The tasks were as follows: firstly, each participant was told to locate and save the bearing of the cast device; once a bearing was set they then had to perform the cast gesture; once the device was connected, they were asked to select a video from the list and perform the play gesture; after pausing and playing the video, turning the volume up and down by performing gestures, they were asked to stop casting and to close the application.

6.1.2 Results

By observing the participants performing the action, it was clear that they had opposite experiences. For example, the first participant performed actions with the device without prompting, she said that this was because the gestures "felt natural". When asked to expand on that comment, she said that the gestures she made were appropriate for the command she wished to issue. She also mentioned that her past experience of the Nintendo Wii influenced her

interaction.

The second participant was slower to use the application in general. By using the cast button's overlay(recommended from Google's Cast design checklist), the participant was able to quickly grasp the concept of casting and connecting to the Chromecast device. However, the rest of the the study slowed down as performing the gestures seemed unnatural for the participant. He made slow and deliberate movements, this meant that the device did not recognise flick gesture he made. Even after prompting and guiding him on how to use the device and describing the task he needed to perform, he still seemed to struggle with performing the gestures.

Both participants were told to tilt the device forward and towards the ChromeCast, the first participant moved around the desk and tapped the smartphone to the Chromecast in order to cast the content.

6.2 Semi-structured interview

To further learn about the users experience with the system and to find their thoughts and feelings, we conducted semi-structured interviews.

6.2.1 Methodology

The three of us sat down in a separate room to the device and I asked each of them individually, questions about their experience. I took note of what they were saying and after I had finished asking questions we freely spoke about the system overall.

6.2.2 Results

The first participant preferred to focus her attention on the device than the TV. she felt that she would also benefit from haptic feedback. Making gestures felt natural to her, she also claimed that making gestures was easy. She did express a concern about the battery life of the device, something I had not considered during implementation. When asked about whether she preferred to use a traditional TV remote or the smartphone, she said that she would use whichever is more convenient at the time. She did say that minimal input would impact her choice of device. For example, she would choose the device that required the least number of buttons presses required to complete a certain task.

The second participant paid more attention to the TV than the smartphone, however he preferred audio feedback than any other type of feedback. Performing the gestures felt unnatural to him, this was obvious during the observations. After using the application and completing the tasks, he still preferred using a traditional TV remote. He also added at the end that he dislikes using modern appliances and isn't comfortable using them.

After one participant brought it up, they both agreed that age was an issue with using not only the application, but with emerging technology.

6.3 Nasa task load index

The Nasa task-load index (TLX) is an assessment tool that measures a participants perceived workload in performing a task[4]. The workload is measured in six different scales to measure each type of demand required to complete the task:

- Mental Demand, how much mental and perceptual activity was required? Was the task easy or demanding, simple or complex?

- Physical Demand, how much physical activity was required? Was the task easy or demanding, slack or strenuous?
- Physical Demand, how much physical activity was required? Was the task easy or demanding, slack or strenuous?
- Temporal Demand, how much time pressure did you feel due to the pace at which the tasks or task elements occurred? Was the pace slow or rapid?
- Overall Performance, How successful were you in performing the task? How satisfied were you with your performance?
- Effort, How hard did you have to work (mentally and physically) to accomplish your level of performance?
- Frustration Level, how irritated, stressed, and annoyed versus content, relaxed, and complacent did you feel during the task?

6.3.1 Methodology

After the interviews and discussion, the participants were asked to complete a template of the task load index. Following this step, they were then asked to complete fifteen pairwise questions as part of the TLX. This step required the participants to choose which was the most demanding out of two different types of scales.

6.3.2 Results

	A	B	C	D	E
1	Subject name:	M			
2	Task text:	The SmartRemote observation			
3	Date/Time:	05/09/2020 19:49			
4					
5	Scale name	Raw Rating	Tally	Weight	Adjusted Rating
6	Mental Demand	60	2	0.133333333	8
7	Physical Demand	10	0	0	0
8	Temporal Demand	50	4	0.266666667	13.33333333
9	Performance	75	3	0.2	15
10	Effort	30	1	0.066666667	2
11	Frustration	50	5	0.333333333	16.66666667
12					
13	Overall rating	55			

Figure 10: TLX from the first participant.

6.4 Analysis And Discussion

The results in this section show two different experiences and preferences from the participants. While having similar backgrounds, their performance and feedback differed when taking part in the study. The data from this study shows some contrasting views among these participants. The results are mostly similar to the results in the research mentioned ealrier.

	A	B	C	D	E
1	Subject name:	J			
2	Task text:	The SmartRemote observation			
3	Date/Time:	05/09/2020 20:07			
4					
5	Scale name	Raw Rating	Tally	Weight	Adjusted Rating
6	Mental Demand	15	2	0.133333333	2
7	Physical Demand	15	4	0.266666667	4
8	Temporal Demand	20	2	0.133333333	2.666666667
9	Performance	25	3	0.2	5
10	Effort	15	3	0.2	3
11	Frustration	15	1	0.066666667	1
12					
13	Overall rating	17.66666667			

Figure 11: TLX from participant two.

7 Discussion and Conclusions

This document presents a literature survey on mobile-spatial interaction; using the findings from the survey, an implementation to control media on a display; and user studies, both before and after implementation. We built on some work from the literature review to develop a prototype that controls the media on a digital display, using a Google Chromecast. The prototype consists of two parts: an Android smartphone and a software application. The application reads the data from the smartphones sensors to recognise gestures. The application recognises gestures through hard coded parameters which form a threshold, the application constantly reads the data from the sensors and when the sensor data falls within the threshold it recognises the gesture. After looking at the literature surrounding the topic, we distributed a questionnaire as part of our data gathering phase. The results from our questionnaire helped give direction for this work. Once we developed the prototypes from the data gathered in previous stages, we performed observations and post-observation interviews. The data from this study allowed us to evaluate our prototype for further development, the data also gave us a further insight into this subject.

7.1 Project Reflection

This project spanned over several months and builds upon a specification document created earlier this year. The state of the prototype at the time that this document is written, satisfies some of the aims and objectives set towards the beginning of the project. The prototype can connect to a Chromecast and display media over a network and onto a digital display, it can also control the content on a basic level. Further work can be done to increase the number and types of devices the application can interact with. Also, more actions could be implemented for each type of device the application is connected to. The prototype is limited to only connecting to a Google Chromecast that is already setup and is on the same network as the application. The application can recognise basic gestures, this includes: tilting and rotating left or right, tilting forwards and flicking towards the user. This is a small vocabulary of gestures and is hard-coded, the prototype and user experience would benefit greatly if the user could create and assign their own gestures. The accuracy of the gesture recognition should be improved before further steps are taken with the prototype, this could be done with Tensorflow Lite, especially since the literature in this document has not used it and it is a new technology.

7.2 Insight into feedback modalities

Our work gave an opportunity to explore how different modalities can be used for feedback to the user. Alongside the findings in the literature shown in this work and findings from our user studies, further investigation into the impact of feedback may be worthwhile. Multiple aspects for this type of interaction are open to exploration, these include: The amount of feedback for the user as too much feedback can be bad/overwhelming[19]; concepts similar to Actioncube where content on the screen of the device can aid the users experience; also, data from our user studies show that different users focus on different parts of the system. Multiple prototypes for the same system could be created to evaluate different types of feedback across various modalities. For example, prototypes for haptic-only or visual-only feedback.

Some time was spent in implementing prototypes that displayed different content on the devices screen. A blank screen, view from the camera, and video content were considered and tested. We decided not to continue with this aspect.

7.3 Limitations

The hardware used in this work is off the shelf commercial products, so were limited to the capabilities that the hardware offered. Using a Raspberry Pi or Arduino as a router would offer more freedom in development, this would also make it easier to write the software that allows the the end devices to connect.

We used a Chromecast to display and control the chosen content on to the digital display, this was a significant limitation as we could only use the Chromecast under its limited functionality. Such as requiring an internet connection to work, this meant that we could not setup our own portable local network for development. Also, Googles Chromecast is one of the IOT devices that is not compatible for use or setup with eduroam. Although Google have published guides and code for developers to implement casting functionality into their apps, the code published is abstract, hidden, and based on dependency injections.

7.4 Future work

This work has not only explored some aspects of current research, but shown some opportunities for further work. A method of supplementing the data from user studies is by using Google Play to publish the app, ratings and reviews would then be used for further evaluation. There is the restriction that the application can only be used for users with a ChromeCast device.

To further develop data from user studies, use of the device at different times of the day could be investigated. For example, how does use of the device differ between using it at the start of the day before work, compared to the evening after a day of work? How does timing of using affect the cognitive load and satisfaction from using the device? Perhaps comparing our implementations to current, traditional implementations(such as a normal TV remote) by performing the same user studies, would yield valuable results? One of the most significant and promising difference between our implementations and the implementations discussed in the literature is gesture recognition. As mentioned previously, each publication that was investigated earlier in this document had a different method of recognising gestures, none of which were Tensorflow lite. It may be promising to investigate the accuracy of Tensorflow lite in recognising gestures compared to other methods, such as Bayesian networks and back-propagation neural networks. It may be useful to make these comparisons as Tensorflow lite is the only implementation that is designed for use with mobile devices.

Recalling the gesture vocabulary of an application has been briefly mentioned in the literature survey of this document however, this element of the area has not been thoroughly

explored. This element may be worth looking into, especially as this is an emerging interaction. Gesture recall was mentioned in our questionnaire, when asked "On a scale of one to five, how confident would you be remembering the different gestures required to control devices?", the responses were spread almost evenly across the scale. It could be worthwhile to implement a feature that allows users to assign their own custom gestures to the commands of a media controller. This would introduce a different approach in recalling the gesture vocabulary as users are remembering gestures that they thought of, rather than remembering a set of default gestures.

Some elements of this type of interaction that has not been explored is usability in certain contexts and accessibility. Alternative modalities for some technologies(such as voice) host numerous benefits over traditional interactions and other alternative modalities. For example, using voice controlled devices such as an Amazon echo can be useful when the user cannot physically perform an action. Such as when a user requires a timer while baking, or for turning the lights on/off when they have their hands full. Sometimes, using this modality can not be the most effective or suitable. For example, at nighttime when the household is asleep and the user needs to be quiet.

An alternative for this would be through physical interaction, for instance, the traditional TV remote. Or in the context of our work, mobile-spatial gestures. Using these types of modalities can an issue for users with disabilities or impairments, this is something that might hold potential for exploring. Our questionnaire briefly touched upon accessibility by asking about mobile device accessories and types of feedback. We asked: "Do you use any accessories for your device that improve its grip and how it is held? For example, a phone case or a pop socket?" with the aim of looking into accessibility, especially since the device is being moved about at different speeds in physical space. Another question we asked with an aim to looking into accessibility was: "Which type of feedback from your device do you prefer?". The responses from this question could lead onto exploring the second-screening aspect of this type of interaction. Further work would explore users experiences in using a smartphone as a universal remote control for smarthome IOT devices, similar to Ambient wall and MagicWand[15].

References

[1] Ana M. Bernardos, Xian Wang, Enrique García, Javier Portillo, and José R. Casar. Using a platform for mobile gesture-based interaction to control smart objects. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, New York, NY, USA, 2013. Association for Computing Machinery.

[2] Andrea Bianchi and Ian Oakley. Designing tangible magnetic appcessories. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, pages 255–258, 2013.

[3] Sung-Jung Cho, Jong Koo Oh, Won-Chul Bang, Wook Chang, Eunseok Choi, Yang Jing, Joonkee Cho, and Dong Yoon Kim. Magic wand: a hand-drawn gesture input device in 3-d space with inertial sensors. In *Ninth International Workshop on Frontiers in Handwriting Recognition*, pages 106–111. IEEE, 2004.

[4] Lacey Colligan, Henry W.W. Potts, Chelsea T. Finn, and Robert A. Sinkin. Cognitive workload changes for nurses transitioning from a legacy system with paper documentation to a commercial electronic health record. *International Journal of Medical Informatics*, 84(7):469 – 476, 2015.

[5] Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2):393 – 405, 1990.

[6] Gabriele Costante, Lorenzo Porzi, Oswald Lanz, Paolo Valigi, and Elisa Ricci. Personalizing a smartwatch-based gesture interface with transfer learning. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 2530–2534. IEEE, 2014.

[7] William T Freeman and Craig D Weissman. Television control by hand gestures. In *Proc. of Intl. Workshop on Automatic Face and Gesture Recognition*, pages 179–183, 1995.

[8] Finn V Jensen et al. *An introduction to Bayesian networks*, volume 210. UCL press London, 1996.

[9] Juha Kela, Panu Korpipää, Jani Mäntyjärvi, Sanna Kallio, Giuseppe Savino, Luca Jozzo, and Sergio Di Marca. Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing*, 10(5):285–299, 2006.

[10] Hark-Joon Kim, Kyung-Ho Jeong, Seon-Kyo Kim, and Tack-Don Han. Ambient wall: Smart wall display interface which can be controlled by simple gesture for smart home. In *SIGGRAPH Asia 2011 Sketches*, pages 1–2. 2011.

[11] Ming Hsiao Ko, Geoff West, Svetha Venkatesh, and Mohan Kumar. Using dynamic time warping for online temporal fusion in multisensor systems. *Information Fusion*, 9(3):370–388, 2008.

[12] Jukka Linjama, Panu Korpipää, Juha Kela, and Tapani Rantakokko. Actioncube: A tangible mobile gesture interaction tutorial. In *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction*, TEI '08, page 169–172, New York, NY, USA, 2008. Association for Computing Machinery.

[13] Lucy Macken and Paul Ginns. Pointing and tracing gestures may enhance anatomy and physiology learning. *Medical Teacher*, 36(7):596–601, 2014.

[14] Cory Myers, Lawrence Rabiner, and Aaron Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(6):623–635, 1980.

[15] Kazushige Ouchi, N Esaka, Y Tamura, M Hirahara, and M Doi. Magic wand: an intuitive gesture remote control for home appliances. In *Proceedings of the 2005 International Conference on Active Media Technology, 2005.(AMT 2005).*, page 274. IEEE, 2005.

[16] Elif Ozkaya, H Erkan Ozkaya, Juanita Roxas, Frank Bryant, and Debbora Whitson. Factors affecting consumer usage of qr codes. *Journal of Direct, Data and Digital Marketing Practice*, 16(3):209–224, 2015.

[17] Gang Pan, Jiahui Wu, Daqing Zhang, Zhaohui Wu, Yingchun Yang, and Shijian Li. Geeair: a universal multimodal remote control device for home appliances. *Personal and Ubiquitous Computing*, 14(8):723–735, 2010.

[18] Cory Robertson and Tim Green. Scanning the potential for using qr codes in the classroom. *TechTrends*, 56(2):11, 2012.

[19] Simon Robinson, Parisa Eslambolchilar, and Matt Jones. Point-to-geoblog: gestures and sensors to support user generated content creation. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 197–206, 2008.

[20] Simon Robinson, Jennifer Pearson, and Matt Jones. Posterpointing: Making paper displays interactive using mobile devices. In *Proceedings of the First African Conference on Human Computer Interaction*, AfriCHI'16, page 170–175, New York, NY, USA, 2016. Association for Computing Machinery.

[21] E. Rukzio, A. Schmidt, and H. Hussmann. Physical posters as gateways to context-aware services for mobile devices. In *Sixth IEEE Workshop on Mobile Computing Systems and Applications*, pages 10–19, 2004.

[22] Thomas Seifried, Michael Haller, Stacey D Scott, Florian Perteneder, Christian Rendl, Daisuke Sakamoto, and Masahiko Inami. Cristal: a collaborative home media and device controller based on a multi-touch display. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 33–40, 2009.

[23] Sung-Jung Cho, Jong Koo Oh, Won-Chul Bang, Wook Chang, Eunseok Choi, Yang Jing, Joonkee Cho, and Dong Yoon Kim. Magic wand: a hand-drawn gesture input device in 3-d space with inertial sensors. In *Ninth International Workshop on Frontiers in Handwriting Recognition*, pages 106–111, 2004.

[24] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10, 2012.

Appendix

Google questionnaire:
https://docs.google.com/forms/d/16m5yEJVuyq9UZoYcLl3T6BgxWL5KB33vYhKMNTrqJVY/edit?usp=sharing

GitHub repositories:
- https://github.com/Nachodood/The-SmartRemote - https://github.com/Nachodood/CastSmartRemote

Google colaboratory notebooks:
https://drive.google.com/drive/folders/11Ln1NoTa70z4TLkBz_D58NH7W2ak83tG?usp=sharing

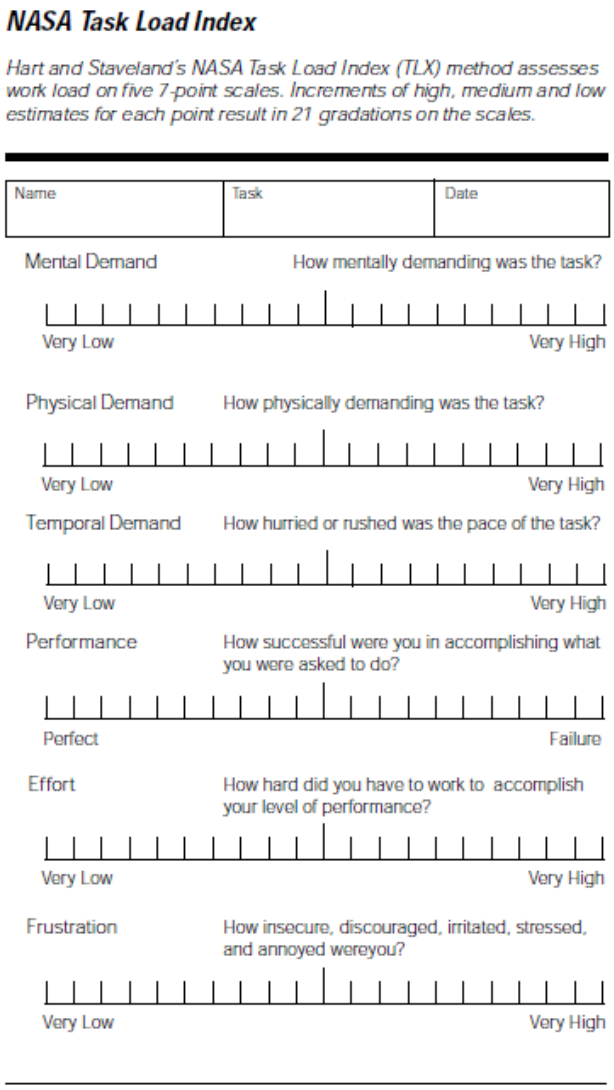


Figure 12: Nasa's task load index.

Record of supervision

RECORD OF SUPERVISION

NB: This sheet must be brought to each supervision and submitted with the completed Dissertation

(to be completed as appropriate by student and supervisor at the end of each supervision session, and initialed by both as being an accurate record. NB it is the student's responsibility to arrange supervision sessions and he/she should bear in mind that staff will not be available at certain times in the summer)

Student Name: Nicholas Cumplido
Student Number: 826488
Dissertation Title: SmartRemote: An implementation and survey of a gesture-based media remote controller
Supervisor: Dr. Simon Robinson

Supervision	***Date, duration***	***Notes***	***Initials Supervisor***	***Initials student***
1: Brief outline of research question and preliminary title (**by pre June**)	26/03/20: 1hr	Project change from hardware to software-based project. Discussed and decided on new project.	SR	N.C.
2: Discussion of detailed plan and bibliography (**by June**)	15/06/20: 1hr 29/06/20: 1hr	Progress update meetings	SR	N.C.
3: Progress report, discussion of draft chapter (**by August**)	13/07/20: 1hr 27/07/20: 1hr	Fortnightly meetings to update on project progress and map out pathway to completion.	SR	N.C.
4: (optional) progress report (**by September)**	10/08/20: 1hr 24/08/20: 1hr	Progress update meetings. Nic gave an update about his implementation and dissertation progress. He has recently started a PGCE course, so is doing both that and the dissertation in parallel. It will be a lot of work, but he has an early draft, and is focusing on refining it to meet the 30th September deadline.	SR	N.C.
5: Submission (**by 30 September**)	28/09/20: 1hr	Review of draft and pointers to key areas that need improvement before submission.	SR	N.C.

Statement of originality

I certify that this dissertation is my own work and that where the work of others has been used in support of arguments or discussion, full and appropriate acknowledgement has been made. I am aware of and understand the University's regulations on plagiarism and unfair practice.

Signed: Nicholas Cumplido Date: 30/09/20

Figure 13: Record of supervision.