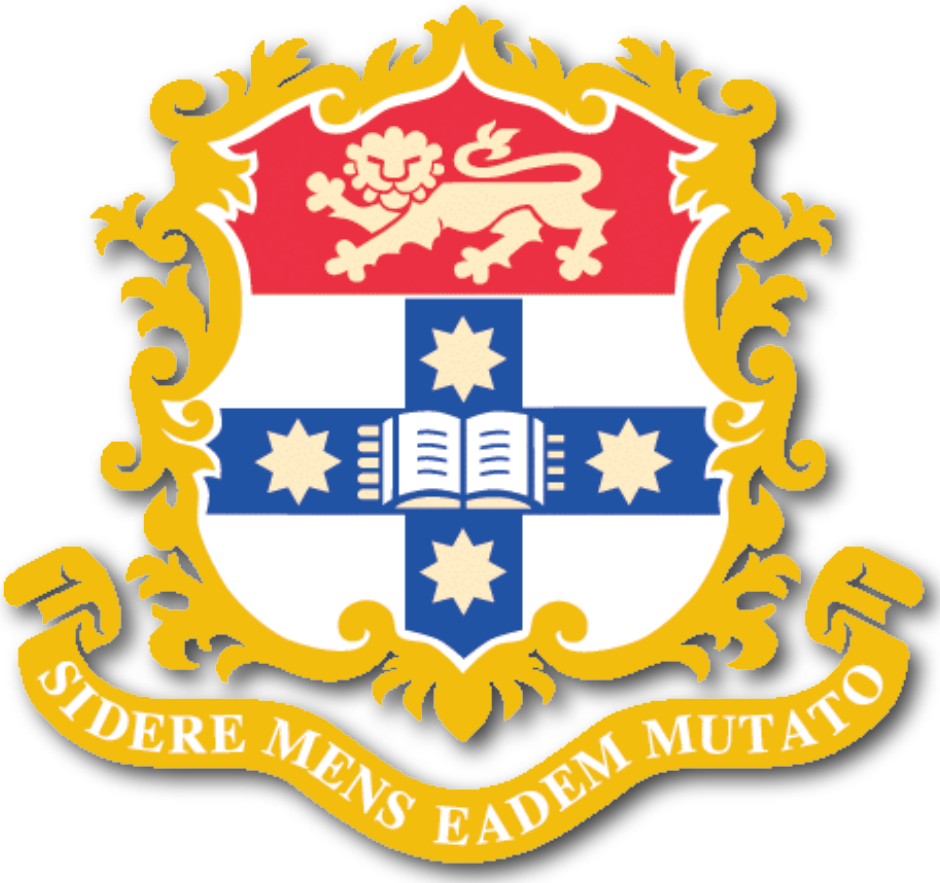
HANDS IN AIR: A WEARABLE SYSTEM FOR DISTRIBUTED COLLABORATION

Jalal Albasri



COMP 5703
Information Technologies Project
Semester 1, 2011

Supervisor: Dr. Zhiyong Wang
Submission Date: 14th June 2011

HandsInAir: A Wearable System for Distributed Collaboration.

HandsInAir is a real-time collaborative wearable system for distributed collaboration. The system was developed to support real world scenarios in which a remote mobile helper could guide a local mobile worker in the completion of a physical task. HandsInAir implements a novel approach in supporting the mobility of both collaborators. The approach taken allowed the helper to perform gestures without the need to touch tangible objects and requiring little environmental support. The system consists of two nodes the helper and worker. The nodes are connected via a wireless network connection and partners communicate with each other via audio and visual links. In this report I review related work, and describe the technical implementation and evaluation of the system and conclude with a brief discussion of future work intended to advance the system.

Acknowledgements.

Dr. Leila Alem, CSIRO ICT Centre, {leila.alem@csiro.au}
Dr. Weidong Huang, CSIRO ICT Centre, {tony.huang@csiro.au}

Table of Contents

Chapter 1. Introduction _____ **1**

1.1 Problem Statement *1*

1.2 Related Work *2*

Chapter 2. System Design _____ **6**

2.1 Background Tools and Concepts *6*

2.2 System Architecture *8*

2.3 User Operations *12*

Chapter 3. Evaluation _____ **14**

3.1 Testing Procedure *14*

3.2 Results *16*

Chapter 4. Process _____ **21**

Chapter 5. Reflection _____ **24**

Chapter 6. Conclusion _____ **28**

References _____ **30**

Appendix 1: Helper Questionnaire *32*

Appendix 2: Worker Questionnaire *34*

Appendix 3: Intended and Actual Schedules *36*

Chapter 1. Introduction

1.1 Problem Statement

Collaboration between individuals across geographic and organisational boundaries has become an essential aspect of our daily lives. In response there has been a growing interest among researchers and engineers in developing systems that support communication and collaboration among remote individuals. The majority of these systems however have been designed to mitigate the collaboration of participants who hold similar and equal roles, such as students working together to complete a group assignment. Relatively less attention has been given to systems in which partners have distinct roles, such as a worker guided by a helper.

As technologies become increasingly complex our dependence on expertise in order to understand and use technology is becoming more apparent. It is a challenge to address this dependency and many real world scenarios exist in which assistance from a remote helper is required to enable a local novice to accomplish a physical or technical task. Within the mining industry, machinery maintenance and repair tasks require specific knowledge and expertise that often have limited local availability. A need exists for a remote engineer with the necessary skills to guide a local non-specialist worker in the process of repairing the machinery. The transferral of knowledge from the offsite helper to the onsite worker would avoid the lengthy and costly process of bringing the expert to the worksite physically. Expert medical knowledge is similarly in limited availability. If the skills required to operate the machinery used to perform an ultrasound scan were not locally available, guidance by a remote radiologist would enable a non-specialist doctor or nurse to produce a quality scan.

It has been shown that a major issue contributing to the ineffectiveness of remote, in contrast to co-located collaboration, is the loss of a common ground through which participants can communicate [13]. Studies have shown that providing participants access to a shared virtual space can be effective at addressing this limitation and beneficial to the completion of collaborative tasks [5, 11]. Tang et al. [17]

characterise shared virtual spaces as "*one where participants can see, share and manipulate artefacts within a bounded space"* and often take the form of a video view of the workspace [15].

Further research indicates that video mediated communication is less efficient than face-to-face communication due to a loss of non-verbal gestures over task objects that would otherwise be visually available to all [11, 12]. A number of systems have been developed to incorporate gesturing into a shared visual space such as the DOVE system of Ou et al. [15], GestureMan of Kuzuoka et al. [14] and HandsOnVideo of Alem et al. [1]. The systems however demand that at least one of the collaborators be confined to a desktop setting and often require a complex technical environment in order to support the sharing of gestures over the collaborative workspace. How to support the communication of hand gestures in a scenario where collaborators are fully mobile and not confined to traditional desktop environments has not yet been fully explored.

For this project I co-developed "HandsInAir", a wearable system for mobile remote guidance. The system implements a novel approach to support the mobility of both remote collaborators. It requires little environmental support and allows the helper to perform gestures without having to touch tangible objects making it ideal for when collaborators are mobile.

1.2 Related Work

When performing a collaborative physical task people interact with each other through various communication channels. Interpersonal communication is most effective when participants share a maximum amount of common ground such as mutual beliefs, knowledge, attitudes, expectations etc. It has been demonstrated that shared visual spaces can be effective in providing a common ground to facilitate communication between participants. In particular, shared views play at least three interrelated roles which are: maintaining situational awareness, aiding conversational grounding and promoting the sense of co-presence [13].

In order to achieve successful collaboration a participant must have ongoing awareness of the state of the current task as well as of their partner. Visual views of the workspace can provide such awareness. Effective communication also depends on how much mutual knowledge partners have about their task and each other. Awareness and mutual knowledge enable participants to develop a mechanism to coordinate verbal utterances and physical actions; what to say and do next. More specifically, as stated by Gergle et al. [7] *"speakers form utterances based on an expectation of what a listener is likely to know and then monitor whether the utterance was understood. In return, listeners have a responsibility to demonstrate their level of understanding"*. The visual information provided in the shared view of the workspace helps build the foundation of mutual knowledge needed for effective collaboration.

The final role shared visual spaces play in mediating collaboration between partners is co-presence. Co-presence helps promote effective collaboration in two ways. Firstly, it provides sources that may be used for grounding [5]; collaborators obtain visual feedback from their partner and task objects can be referred to using a common reference. Secondly, it helps give collaborators a sense of being "together" in the virtual space allowing them to feel more connected even though they are physically distributed. It has long been acknowledged that the reason why face-to-face communication is generally more effective than video-mediated communication is because in the face-to-face condition, participants are able to perform gestures on the task objects and those gestures are visually available to all participants [10, 11].

Although it is not feasible to provide all visual information available to co-located collaborators, to remote collaborators due to bandwidth limitations and the limited cognitive capabilities of humans [5, 13], a shared visual space is instrumental in establishing a common ground between collaboration partners. It is therefore important to determine what visual information is the most helpful and ensure that the information is communicated in an appropriate way when developing systems for remote collaboration.

Fussell and her colleagues conducted a series of studies on collaborative work and found that not only speech, but also gestures and actions were used for grounding and that their use improved task performance [5, 13, 15]. It has also been shown that in role driven remote collaboration with a shared visual space helpers allocate most of their attention to the worker's hands and task objects [7,10]. This has led to the recognition that it is important to support remote gestures when developing tools for remote collaboration.

Two studies conducted by Fussell et al. [6] investigated the relationship between two forms of gesturing, pointing and representational gestures, and how they could be effectively communicated to the remote site. The system used in the first study was mouse based and provided only pointing gestures, while the system used in the second study was pen based and provided both pointing and representational gestures through annotation. The results indicated that simple cursor pointing was not enough to facilitate effective remote collaboration, while pen-based drawing of representational gestures achieved communication and performance comparable to that of co-located collaboration.

An ethnographic study of the nature and role of gestural actions based on a mixed ecology system was conducted by Kirk et al. [9]. The system projected unmediated hand gestures onto the worksite as a way to achieve collaborative awareness. The analysis revealed a collection of 'gestural phrases' that were used for collaborative physical tasks. The gestural phrases consisted of flashing hands, wavering hands, mimicking hands, inhabited hands, negating hands and parked hands.

There have been a number of systems proposed or developed that provide a shared visual space as well as support remote gestures. The WACL system developed by Kurata et al. [16] mounts a steerable camera and laser pointer on the worker's head. It allows the helper to independently set the view of the workspace as well as project a laser spot to point at real objects in the workspace. Similarly the GestureMan systems developed by Kuzuoka et al. mount a camera and laser pointer on a robot located at the worksite. The joystick controlled mobile robot also allowed the helper

to independently set their view of the workspace and communicate gestures directly onto the task objects with a laser spot.

The DOVE (Drawing Over Video Environment) system developed by Ou et al. [15] integrates the helper's gestures into a live video of the worker's workspace. It allowed the helper to perform gestures over the video stream of the work environment while providing verbal instructions. Sketching and annotations were used to support both pointing and representational gestures.

Kirk and Fraser [11, 12] presented a mixed ecology system that supported remote gestures using unmediated representations of the helper's hands. The helper's hands are captured by a video camera and projected directly onto the remote workspace to facilitate mutual awareness between the collaborators.

Most of these early systems for remote collaboration either assume that the workspace is confined to a fixed desktop setting or support only limited gestures such as pointing. The HandsOnVideo system developed by Alem et al. seeks to support real world scenarios in which the worker is completely mobile and in a non-traditional-desktop environment. The helper's hands are captured by a video camera and integrated with a video of the workspace. The combination of the helper's hands and the workspace are available to the mobile worker on a near-eye display located just above the worker's eyes.

Although mobility of the worker has been explored, all of the previously mentioned systems confine the helper to a fixed desktop setting in which they are required to use touch or control physical objects to be able to perform gestures. Capturing of hand gestures and conveyance to the worksite often requires considerable technical and environmental support. HandsInAir seeks to achieve mobility of the helper allowing them to perform their gestures free from environmental constraints and with minimal technical support.

Chapter 2. System Design

2.1 Background Tools and Concepts

The system is comprised of two distributed nodes and collaboration takes place in a directed role oriented manner. The two roles are the worker who is present at the worksite and the helper or instructor who is removed from the worksite.

Figure 1: helmet with mounted peripherals

The Hardware for the system was developed to be identical at both nodes to allow easy swapping of roles if necessary. Hardware consisted of a Microsoft Lifecam webcam mounted on top of the brim, and a Vuzix 920 Wrap near-eye display mounted beneath the brim of the helmet. The webcam is used at the worker's node to capture the view of the worksite and at the helper's node to capture hand gestures performed in the space directly in front of the helper. The system combines the hand gestures with the live video feed of the worksite and the combined view is displayed to both collaborators via the near-eye display. A microphone headset is used to implement an audio link between the participants to facilitate verbal communication. The peripheral devices are connected to a net-book worn by the participants in a backpack. The net-books used had 1.6Ghz Intel Atom processors, 1GB of RAM and ran Windows XP. The net-books were chosen for their low weight and size which allowed them to be easily worn by users.

The system was developed in C++ with Microsoft Visual Studio 2010 on Windows XP machines. A number of external libraries were utilised to perform networking and computer vision operations. C++ was chosen as the development language because of the system's high performance requirements, familiarity of the developer and compatibility with external libraries such as OpenCV and libjpeg.

OpenCV is an open source library containing over 2000 computer vision and image manipulation functions. It was chosen because of it's large range of functions and flexibility and was used to implement simple image manipulations, windowing, display and capturing frames from the camera as well as colour hue filtration for hand gesture recognition.

Due to network bandwidth limitations and the system's real-time latency requirement raw camera frames were too large to be transmitted over the wireless network unless they were compressed first. The system uses libjpeg-turbo, an optimised derivative of the open source IJG JPEG image compression library libjpeg. The standard IJG libjpeg implementation was initially used for image compression, however the performance of the compression and decompression functions was too slow. Libjpeg-turbo accelerates baseline JPEG compression and decompression for up to a 100% performance boost and enabled the system to achieve its real-time frame rate. Although OpenCV has built in JPEG compression and decompression algorithms they are restricted to the reading and writing of images from and to disk and not memory.

Microsoft technologies were used to support many of the HandsInAir functions. Winsock 2 (Windows Sockets) was used to facilitate network communications. This allowed HandsInAir to exchange data independent of the network implementation between the two nodes. Microsoft Foundation Class (MFC) Library was utilised to multithread the core functions of the HandsInAir system enabling them to run concurrently in an event driven fashion. Events such as capturing a frame from a camera or receiving a frame from a socket triggered reactive activities, such as sending or displaying the frame on the near-eye display.

2.2 System Architecture

A SharedBuffer class was used to enable the exchange of frames between threads. Within the SharedBuffer class frames were stored in a simple character array. Two simple functions were implemented to read from and write to the SharedBuffer. Critical sections were used to lock the buffers during read and write operations so that only one thread could access or modify the image in the buffer at any one time. The SharedBuffer objects were held as global variables so that they could be accessed by all threads of the HandsInAir program.

The HandsInAir program at the helper node began by launching it's four major functions h_Camera, h_Send, h_Receive and h_Display, each function is started in it's own thread. Two SharedBuffer objects, SendBuffer and DisplayBuffer, were used to exchange frames between the threads. Two Event objects, Send and Display, were used by threads to signal the completion of tasks and enabled them to synchronise their operations. An additional Quit event was used to signal the receipt of a quit command from the user and end the program. The most significant operations of the helper program can be seen in the activity diagram below.

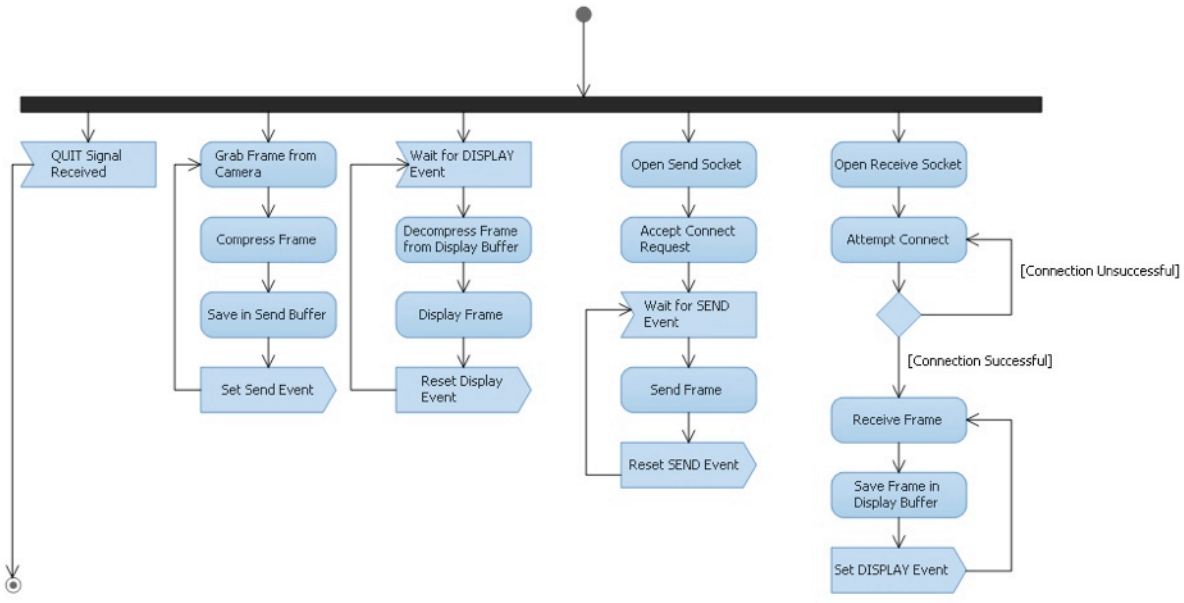


Figure 2: helper node activity diagram.

The h_Camera thread's purpose is to continually update the SendBuffer with new frames from the local camera. It consists of a continuous loop that queries the camera for a new frame, and upon receiving the frame compresses it to a JPEG and saves it to the SendBuffer. Finally the h_Camera thread signals the Send Event indicating to the h_Send thread that the buffer has been updated with a new frame that is ready to be transmitted.

The h_Send thread begins by setting up a Winsock socket upon which it listens for an incoming connection. The function then waits on the listen socket for the remote node to attempt a connection. If a connection attempt is detected it is accepted and a second Winsock socket is created and used to send the image data. The h_Send function then enters a continuous loop that begins by waiting for the Send event to be signalled. Once the Send event is signalled by the h_Camera function, h_Send knows that there is a new frame in the SendBuffer that is ready to be sent. h_Send reads the image out of the buffer, sends it to the remote node over the socket connection and finally resets the Send event. It then returns to the top of the loop where it waits to the Send event to be signalled by the capturing of another new frame.

Just as the h_Send and h_Camera functions coordinate their actions to accomplish the goal of sending a frame, the h_Receive and h_Display functions synchronise their actions in order to receive and display a frame from the remote node. h_Receive begins by setting up a receiving socket and attempting a connection to the remote node. Upon the successful establishment of a connection it enters a continuous loop in which it receives a frame from the socket and saves it in the DisplayBuffer. At the end of the loop it signals the Display event to notify the h_Display thread that a new frame has been received and is ready to be displayed.

The h_Display thread similarly operates a continuous loop, at the start of which it waits to be signalled by h_Receive through the Display event. Once it has been notified of the arrival of a new frame it reads the frame out of the DisplayBuffer and decompresses it from a JPEG into OpenCv's IplImage format. h_Display then outputs the frame to the user by updating a window on the near-eye display and resets the Display event.

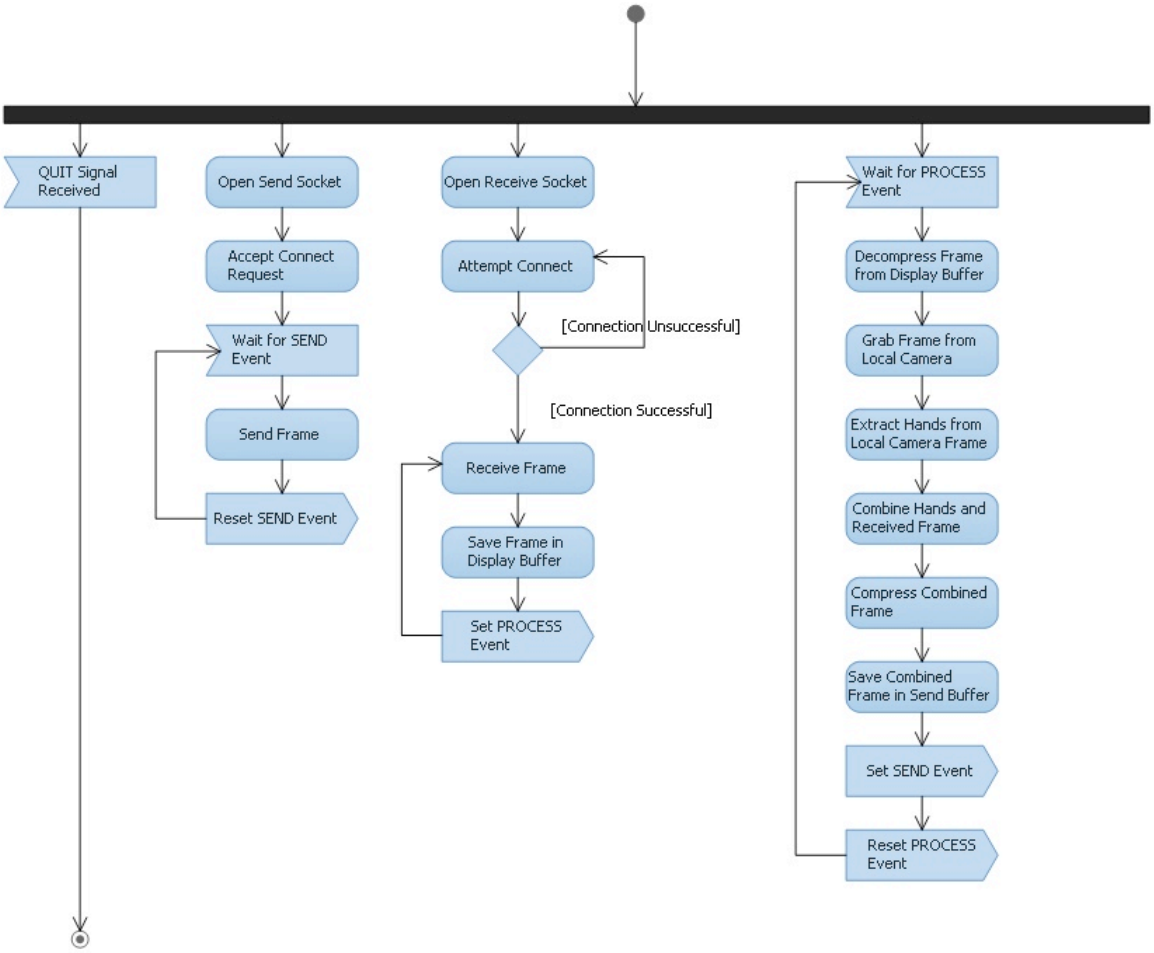


figure 3: worker node activity diagram.

The HandsInAir program at the worker node is comprised of three major functions w_Send, w_Receive and w_Process. They are all started by the main function in separate threads in a similar fashion to the helper program's functions. Two Event objects, Send and Process, are used to synchronise the threads and two SharedBuffer objects, SendBuffer and DisplayBuffer, are used to exchange image frames between the threads. A Quit event is used to signal termination of the program in the same way as the helper program.

The operations of the w_Receive and w_Send functions are similar to their counterparts in the helper program. The w_Receive function establishes a connection to the helper node and receives a frame containing the helper's hand gestures over an arbitrary background. It saves the frame in the DisplayBuffer and signals the Process event.

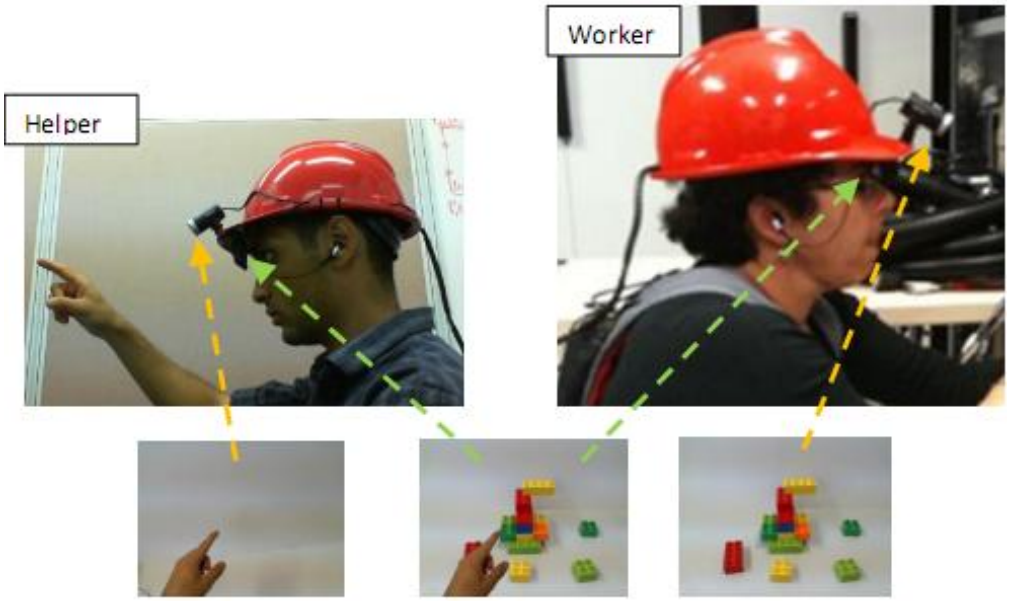


Figure 4: camera captures and near-eye display views

The w_Process function is where the majority of the program's activity takes place. It operates a continuous loop that waits on the Process event to be signalled by the arrival of a new frame from the remote node. It then reads the frame from the DisplayBuffer and decompresses it from a JPEG to OpenCV's IplImage format. Next w_Process uses OpenCV functions to extract the hand gestures from the received image, and overlay them onto a new frame of the worksite captured by the worker's local camera. The combined frame is displayed to the worker by updating a window on their near-eye display, then compressed to JPEG format and saved in the SendBuffer object. Finally the w_Process function signals the Send event to indicate there is a new frame ready to be sent and resets the Process event.

Extraction of hand gestures from an arbitrary background in the frame received from the helper node was originally achieved using the OpenCV AdaptiveSkinDetector algorithm developed by Dadgostar et al. [25]. The algorithm's skin tone detection is based on expected hue and saturation values of skin. A histogram of expected HSV values was built by manually segmenting a set of 20 training images. The histogram is not only used to filter skin values from the input image but adjusted on the fly with every subsequent frame to home in on the skin tone actually appearing in the image. Although the Dadgostar algorithm was quick and worked reasonably well in controlled conditions, significant differences between frames, poor lighting conditions and image compression quickly degraded the robustness of the skin tone

detection and resulted in regions of skin not being detected as well as background artefacts being falsely detected as skin. The requirements called for high robustness in hand gesture detection not only to be able to support the helper in varying environmental conditions but also to convey hand gestures to the worker as accurately and clearly as possible. The adaptive skin detection algorithm was replaced by requiring the helper to wear a pair of blue gloves, which enabled highly robust and efficient hand gesture recognition with simple colour hue filtration of each frame. There was a concern that replacing natural hands by gloves would result in a loss in the richness of the information conveyed by the hand gestures and so two toned blue gloves were chosen so that their orientation would be as clear as possible to the worker.

Figure 5: gloves used for detection

2.3 User Operations

The HandsInAir system is designed to enable user's to collaborate in a distributed environment by interacting with a physical worksite or virtual image. The user's are comprised of the worker at one node and the helper at the other. The helper is able to instruct the worker by performing hand gestures over virtual objects at the worksite displayed on their near-eye display. The worker capture's the worksite using the helmet mounted camera and is delivered instructions through hand gestures over worksite objects displayed on their near-eye display. Both the helper and the worker can communicate verbally over an audio link.

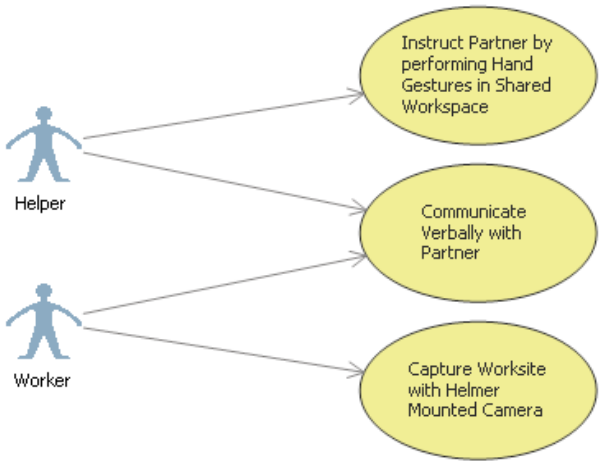


Figure 6: use case diagram.

Neither a user interaction interface or direct interaction mechanism with the program is required allowing the worker to maintain unconstrained interaction with the worksite and task objects and the helper free to perform hand gestures in front of the camera.

Chapter 3. Evaluation

3.1 Testing Procedure

A usability study was performed to evaluate the HandsInAir system. The study was designed to assess the systems capabilities at facilitating distributed role oriented collaboration as well as to be a test of the concepts underlying the use of hand gestures to mediate communication.

Testing of the system was conducted at the CSIRO ICT Centre. A workshop was used to simulate the worker's environment and an office was used to simulate the helper's environment. A wired network connection was laid between the two rooms,and a wireless router was used at each end to provide the systems with wireless connectivity throughout the testing environment, allowing the users to experience unmitigated mobility with the equipment.

Figure 7: Lego bricks at worker station 1

To test the systems' capabilities at facilitating remote collaboration of physical tasks, users were asked to work together to build simple shapes with Lego bricks. Two stations were set up at the worker's site, each station having a number of lego bricks scattered randomly. At the start of the test the helper would instruct to worker to build a letter of the alphabet using the lego bricks at the first station. In order to test the worker's mobility the helper would then ask the worker to put the model down and move across the room to the second station where they would carry out a similar task with another set of Lego bricks. A number of obstacles were placed between the two stations in order to assess the worker's awareness of their physical surroundings while wearing the head gear. The worker would then be asked to return to the first station and combine both shapes. In order to explore the mobility at the helper site participants were asked to deliver the instructions to build the first shape from a seated position and for the second shape from a standing position. When instructing

the worker to combine the two shapes in the third task the helper was asked to deliver the instructions from whichever position they preferred, seated or standing.

Participants were procured from the CSIRO ICT Centre and were equally comprised of researchers and students. Almost all participants had a background in human computer interaction and were able to use their knowledge of the field to provide insight into the strengths and weaknesses of the system. There were 10 participants in total, tests were conducted in pairs with one partner playing the role of the worker and the other the role of the helper. At the end of the test they were asked to fill out the first questionnaire. They then switched roles and carried out the tasks a second time. A final questionnaire was then administered, followed by a debrief session.

Figure 8: worker carrying out assembly task

The testing criteria included determining whether the system was easy and intuitive to use and if the users found it enjoyable to communicate in such a manner. From the helper's perspective it was desired to determine their experience guiding their partner using pointing gestures as well as representational gestures such as communicating assembly instructions by making shapes with their hands.

The questionnaire was designed to collect qualitative data about the users' experiences with the system. Additional data was gathered through still and video capture of both collaborators using the system during tests as well as interviewing the participants post-testing. Sample helper and worker questionnaires are included in Appendices 1 and 2 respectively. Questions focused on the systems usability as well as the users' sense of co-location with their partner and awareness of their physical environment.

3.2 Results

Participants felt strongly that the system was intuitive to use and easy to get accustomed to. They generally expressed a high level of satisfaction with their task performance and with the extent of the communication with their partners. Tasks were completed with ease and speed. Hand gestures were regarded as extremely useful to communicating task actions however their primary form of communication was always verbal.

Figure 9: helper performing pointing gesture

Difficulty determining how the Lego bricks could be manipulated and what shapes could be created was observed at the helper's side when participants began the unfamiliar task. Once the task was underway however the helper's confidence grew with their increased understanding of the task. This difficulty was not observed in participants who assumed the role of the helper directly after having performed the role of the worker. Their familiarity with the task allowed them to be more confident in their instructions and deliver them quicker and with greater ease. As expected all participants performed and communicated better the second time they collaborated on the tasks.

The majority of participants indicated that they did not feel as if they were at the same location as their partners. This was more obvious in the case of the helpers who indicated they felt removed from the worksite. I believe it is related that many of the participants playing the role of the helper felt strange being able to gesture over objects in the workspace yet unable to grasp or manipulate them directly. A number of participants attempted to compensate and appropriate the task objects by approaching either the table-top or the wall and touching their surfaces when

gesturing over task objects in the virtual workspace. This indicated some participants felt a discomfort with the lack of tangible objects to facilitate gesturing.

Participants expressed no preference gesturing while in a seated or a standing position while in the role of the helper, with an equal number preferring one over the other and some indicating no preference at all. When asked to assume the position they preferred to complete the final task all participants remained in the position they were in for the previous task. This may indicate that the participants felt restricted in movement by the system when playing the role of the helper. In contrast all participants performing the role of the worker found it very easy to move around the worksite while wearing the system. All worker's were able to navigate around obstacles in the room with extreme ease indicating a solid awareness of their physical environment.

Helpers were asked to use both pointing gestures as well as complex representational gestures to demonstrate assembly instructions to their partner. Although they reported equal ease in using both gestures their partner's found it easier to understand the pointing gestures over the complex representational gestures. All participants agreed that pointing was more constructive to the task than showing shapes or assembly gestures. Many of the helpers reported difficulty perceiving the depth of lego bricks as well as greater ease indicating flat assembly instructions as opposed to vertical ones. A lack of depth in the two dimensional image would contribute to the participants' preference towards pointing over representational gestures. Difficulty was also observed when helper's were unable to judge the thickness of some lego bricks. This resulted in clarification required from the worker to amend an incorrect assembly instruction.

The relationship between participants was observed to play a role in the quality of their interactions. Some participants found the role of the worker rigid and confined to only following instructions. Others would take a more collaborative approach to the task, suggesting and trialling configurations without the explicit instruction of the helper. These collaborative workers were observed to use a trial and error strategy

when they were unsure about an instruction. The worker would try a configuration and hold it up to the camera to get the helpers approval.

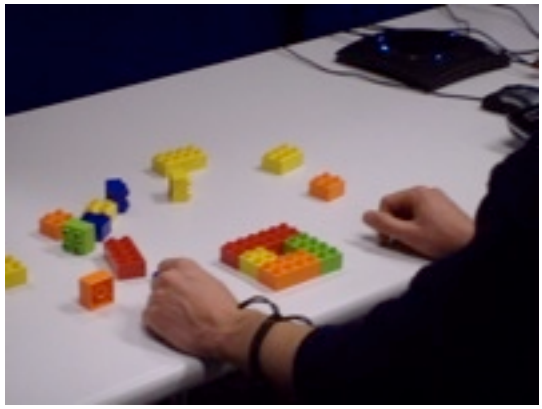


Figure 10: worker receiving instruction

Participants in the role of the worker reported they had difficulty performing task actions and receiving instructions simultaneously. When a helper would administer instructions while the worker was performing an assembly task they would easily get confused as their attention would be split between the physical task and the near-eye display. Furthermore many participants expressed a sense of clutter on the near-eye display when four hands would appear at the same time. Almost all participants began innately coping with these difficulties by staggering the instructions and the execution of assembly tasks reinforcing the observations of Gergle et al. [7]. Worker's were observed to remove their hands from the view of the worksite while receiving instructions. They would then carry out the assembly task, place the object down on the workbench or hold it up to the camera and await confirmation from their partner that they had carried out the assembly successfully. Similarly helpers were observed to promptly remove their hands from the view after delivering their instructions and observe their partner carry out the assembly task without interference. All participants agreed that the system would be most suitable to non-time-critical tasks because of the slow mechanism of interaction. Unexpectedly however is that most participants did not find this cumbersome but more natural.

As mentioned, participants' performance increased with greater familiarity with the task. Two out of the ten participants, having previously played the role of the helper, were observed to sort the Lego bricks by colour and size prior the start of the test, when they played the role of the worker. The majority of participants found it much easier to carry out tasks as the worker than to give instructions as the helper. However when asked which role they favoured participants were equally divided

between the two roles. Participants who favoured the helper expressed they found the role more enjoyable because they felt greater control even though the majority of the responsibility and ambiguity resided on the helper's side. It was indicated that the greatest mental demand came from structuring interactions within the limitations of the system as well as synchronising instructions with their partner. This was best expressed in the words of one participant who said that *"interacting with the system is easy, learning what it is useful for is where the curve is"*.

Participants generally expressed an overall comfort with the equipment. They found it easy to use the near-eye display while maintaining awareness of their physical environment and intuitive to interact with the system in the intended manner. One participant found the experience challenging because the helmet would not fit correctly on their head. The participant had to tilt the helmet forward far enough for the near-eye display to be visible. This however resulted in the camera mounted on the brim becoming aimed almost directly downward. The situation would have been improved if the camera and near-eye display were independently adjustable however the nature of their current mounting on the helmet prevented this.

The majority of participants indicated no hindrance or discomfort by a lag or delay. One participant felt strongly while playing the role of the worker that the there was a large discrepancy between what was displayed on the near-eye display and what could be seen in real life due to a lag which induced a sense of nausea. The same participant experienced difficulty correlating hand gestures on the eye-display to task objects in the physical workspace. Sensing their partner's difficulty and discomfort the helper began administering instructions almost entirely verbally. As they continued the worker was observed to take over control of the task with the helper only consulted for approval of an assembly decision once it had been made. The same participant in the worker role also expressed feelings of "*claustrophobia*" while wearing the head gear and felt their awareness of their physical surroundings had been greatly diminished. The participant however had no difficulty manoeuvring around obstacles in the room.

I had expected to hear from participants that they found it difficult to coordinate holding the worker's view still while the helper gestured over task objects. This feedback however was only received from a single participant. This could have been due to the inherent mechanism of communication that was observed emerge from the remote interactions where the worker would hold their head and hands still while receiving an instruction. Some participants expressed that they found the tasks too easy and that they did not require true collaboration. A few of the participants likened the experience to that of playing first person video games and felt that prior experience with games made adjustment to the equipment easier.

One of the participants experienced difficulty adjusting to the audio link. Unlike a normal phone the audio link between participants incorporated a silence detection feature. The feature would disable the audio link when it detected no users were speaking. When it would detect a spike in the volume the audio channel was reopened. Although the feature worked well the participant found it difficult to determine whether there was someone on the other end because when no-one would talk they could hear total silence.

The system was successful in mediating the remote collaboration. It showed value providing a shared workspace as a basis for common ground between collaborators. All pairs were able to complete the required tasks with relative ease and many found it enjoyable. The system worked very well communicating pointing gestures in the shared workspace. All participants however cited low image resolution as the biggest weakness of the system and found that objects often had to held up to the camera or clarified verbally to be correctly perceived from the helper side. Furthermore many participants found that the lack of depth perception on the two-dimensional image to be the system most limiting factor. Participants were observed to exhibit difficulty from the helper's perspective in discerning the exact sizes and positions of bricks in the workspace as well as communicating complex representational hand gestures.

Chapter 4. Process

The HandsInAir system is the latest evolution of multiple predecessor systems for remote collaboration developed by the CSIRO. The most recent of these systems if HandsOnVideo developed by Alem et al. [1]. HandsOnVideo similarly used pointing and representational gestures overlaid on a live video of the worksite to facilitate a shared workspace for the distributed collaborators. The HandsOnVideo system implemented the helper's station as a tabletop touchscreen upon which the worksite feed from the worker's camera was displayed. A camera was mounted over the tabletop to capture the helper's hands as they gestured over task objects in the shared workspace. From a worker's perspective the systems are identical however HandsOnVideo confined the helper to a desktop environment that required significant technical support to implement.

The HandsInAir project's primary goal was to enable the helper to posses the same mobility and independence from environmental support as the worker. The project began in early February 2011 and was scheduled to continue to end of June 2011. The HandsOnVideo system used proprietary software to establish its connection between the two nodes, exchange data and display the workspace. The CSIRO required a novel software solution to be developed for the HandsInAir system that would free them from dependence on third party solutions. Major milestones for my project included development of the software to run the system, building of the prototype mobile helper system and deployment of the software on said system. The project would culminate in the conducting of a usability study to test both the capabilities of the system developed at fulfilling it's intended purpose as well as the concepts underlying remote collaboration systems. Additional work was scheduled to explore enhancement or evolution of the system. Most notably the CSIRO wished to explore ways in which three dimensional viewing of the workspace could be incorporated into the HandsInAir system. An independent project at the CSIRO had developed a system for real time three dimensional scanning of a workspace and transmittal of an image as well as a depth map of the space over a network.

The project began with major work on the development of the software to run the system. I employed a software development lifecycle that consisted of iterative phases of analysis, design and implementation. Each phase would be ended with a 'quick and dirty' test of the system between myself and colleagues. Once the system was ready a usability study was planned to discover the strengths and weaknesses of the system from the perspective of a user. Additional explorative work on the incorporation of a three dimensional workspace was scheduled for after the usability study until the end of my time on the project.

The design and implementation phases consumed more time than initially expected. This resulted in a delay of the usability testing of almost four weeks. Time available at the end of the project to explore the intended enhancements was reduced. Intended and actual schedules of the major tasks and their time requirements are available in Appendix 3.

Multiple factors contributed to the delays experienced in development. Firstly the work required to build the software solution was underestimated. Secondly flawed design choices early in development were made regarding software and hardware technologies to be used. This resulted in the third factor which was having the wrong tools to carry out the task.

I began the project with limited multimedia or computer vision programming experience, and no networking or multithreaded programming experience at all. The requirements of the application called for the time-critical exchange and manipulation of data leaving little room for delay. I had much to learn about the different technologies available to fulfil my requirements and the early stages of the project were spent familiarising myself with these new solutions. Furthermore I was also the only person on the team with any programming experience and was unfamiliar with development in a Windows environment, as such it was very difficult to judge exactly how much time development would require.

Design and technology decisions were made early on in the project before the requirements were fully understood. OpenCV was used to capture frames from the

camera which limited the maximum attainable resolution to 640x480. Hardware was chosen before the start of the project before software decisions were made. The net-books used to implement the system were chosen because of their relatively low weight and cost. They however did not have sufficient graphics processing or network performance to enable the development of a more cutting-edge system. Skin tone detection and networking algorithms that were developed and showed satisfactory performance on a desktop computer would not when run on the net-books forcing further compromise and re-factoring of code.

The CSIRO was interested to attempt an integration between the HandsInAir system and a separate system developed for three-dimensional mapping of a workspace. The three-dimensional mapping system used JPEGs to transmit it's images of the worksite. In the interest of promoting the possibilities of integration between the two systems JPEG was chosen as the compression method of images for transport over the network.

The net-books exhibited extremely low performance in operations such as JPEG image compression and decompression. This severely limited the maximum attainable frame rate and contributed large delays in the feed. To maintain the realtime requirements of the system the resolution and quality of the JPEG compression had to be decreased. The degradation in image quality severely affected the performance of the skin tone detection algorithm's robustness and it began detecting false positives in the background as well as not detecting the skin tone of the hands. The skin tone detection had to be abandoned and a simpler method using gloves and colour hue filtration was used. Modifications such as these took time and effort that would not have been required had the correct computers or compression algorithms been selected for the job.

Strategies for managing the delays experienced in development were two fold. In order to preserve the integrity of the HandsInAir system it's original scope was maintained. This required a augmentation in the scope of the exploratory work of integrating the HandsInAir with the three dimensional mapping system and extension of the schedule by a further week.

Chapter 5. Reflection

Development of HandsInAir was equally challenging and exciting. The software development project drew on a multitude of disciplines including remote collaboration, wearable computing, networking, computer vision, multimedia compression and multi-threaded programming all of which I previously had limited exposure to. I found the project additionally attractive because I had never developed in a Windows environment before. Moreover I was given the opportunity to work with interesting hardware such as the Vuzix near-eye display.

Although I had adequate knowledge of networking principles and protocols before beginning this project, I had never implemented any networking operations in past programming experiences. Selecting the correct networking paradigm and coupling it with image compression and decompression to meet my real-time latency requirements proved the most challenging aspect of the system's development. The first weeks of development were spent conducting research into networking approaches and technologies to extract the best fitting solution to my problem. Networking support provided by the Microsoft Windows platform's Winsock technologies and their related documentation were instrumental in helping me overcome this challenge and implement an effective and efficient solution to fulfil my networking requirements.

Our team had difficulty estimating the time and effort that would be required to build the software system. I was the only team member with programming knowledge and it did not extend over the concerned disciplines. Many early solutions developed required extensive re-factoring. The first major iteration of the solution was developed with no support for concurrency of activities and used blocking sockets to send and receive data. Blocking sockets do not allow to program to execute while a transfer is in progress. These factors resulted in the program being extremely slow. At this stage I had limited knowledge of socket programming or experience with multi-threaded programming to easily diagnose and rectify the code. It took a redesign of the system to discover my faults and achieve satisfactory performance.

The development process was challenged further by changing requirements. I had originally spent much time researching suitable compression algorithms such as ffmpeg and H.264. JPEG compression was recommended by a senior developer at the CSIRO not affiliated with the project. Furthermore because using JPEG encoding to compress the images offered a potentially simplified integration with the three dimensional mapping system it was chosen as the method to use. The low performance of JPEG compression and decompression however reduced the quality and latency achievable by the system. I was able to increase the performance of the algorithm by replacing the stock IJG implementation of the JPEG algorithm with an accelerated version called libjpeg-turbo, however at this stage in development it was deemed to late to replace JPEG with a more suitable algorithm for video such as H.264. The degradation in image quality disrupted the skin tone detection algorithms in use and a colour hue filtration strategy had to be used to detect hand gestures instead.

OpenCV's primary function in the system was to implement the adaptive skin detection algorithm. As OpenCV was already being used for the hand extraction it was also used for simple image manipulations, windowing, display and capturing frames from the camera as well. OpenCV however would only allow the capturing of frames from the camera at a resolution of 640x480 limiting the quality of the video feed. Even though the original skin tone detection algorithm was eventually replaced the limitations of OpenCV in the program persisted.

This project was an opportunity for growth not only with challenging programming but in other areas as well. A marked difficulty I faced was communicating technical complications, requirements and progress to my fellow team-mates who possessed no programming background. Senior developers who offered their advise in the early stages of the project were often unfamiliar with the system's background or requirements. Time was lost investigating recommended solutions to problems that would not be implementable on the platform or clashed with requirements of the system. As my understanding of the subject matter grew, my confidence in collaborating with other developers strengthened and I was able to conduct much

more instructive consultations with developers from the CSIRO ICT Centre. This was instrumental to the success of the project.

Managing my commitment to other university courses as well as devoting enough time to the project was challenging in the later stages of development and towards the testing phases. I had to adhere to a strict schedule to ensure all my work was completed satisfactorily and delivered on time.

The project was an extremely valuable opportunity to work with a leading ICT research centre as well as to be part of a dynamic and motivated team. This was my first experience working with a team in an work scenario and I found it markedly different from working with a group in a university project. I learnt that confidence was required to assert ideas in a team environment. Furthermore as my knowledge of the disciplines associated with building the system grew I learnt to trust my judgements and question others more thoroughly.

I gained first hand experience with a quick paced software development cycle and learnt how to dynamically manage my development schedule to react to changing requirements and setbacks in development or testing. I recognise that more time was required to have been spent extensively researching the impacts and implications of design decisions, and had this been done more thoroughly time and effort spent refactoring code could have been avoided.

The project also gave me an opportunity to design and carry out my own usability study. I gained insight into the difficulties associated with questionnaire design and administration as well as managing participants and making acute observations during trials.

In retrospect I feel strongly that I could have made better use of resources available to me from within the CSIRO ICT Centre. I learnt however that the right time to ask for help from other developers was after having researched possible solutions in depth myself. When advise was sought too early on I found difficulty communicating

the details of the problem and understanding proposed solutions. Conversely when advise was sought too late time was lost exploring ill-suited solutions.

I do not however feel that time spent exploring solutions that were not suitable for the final product was entirely misspent as all research expanded my knowledge in the concerned disciplines. I found solving networking and multimedia compression problems exceptionally rewarding, and have gained invaluable understanding in these fields that will be significant in my future as a software developer. I feel strongly that the most important skill I developed over the course of the project was the capacity to teach myself about new topics, technologies and techniques in computer programming and build unique solutions to unfamiliar problems.

Chapter 6. Conclusion

HandsInAir is a new real-time wearable system for remote collaboration. It employs novel approaches at supporting the mobility of remote collaborators capturing remote gestures. The system enabled the helper to perform hand gestures in the air without the need to interact with tangible objects. The system is lightweight, easy to set up, intuitive to use and requires little environmental or technical support. HandsInAir has demonstrated great capability at mediating remote collaboration and has significant potential for implementation in a wide range of real world applications such as telemedicine or remote maintenance and repair.

The system's greatest strength is the extent to which it was capable at facilitating remote collaboration tasks. In a usability study of the system all participants were able to collaborate effectively to successfully complete a series of remote tasks. A majority of participants expressed comfort and ease using the system, and found it valuable for remote collaboration.

Findings in the usability study conducted further corroborated concepts underlying remote collaboration and finding is previous studies. These included the prevalence of pointing gestures over complex representational gestures, and the value of a shared workspace at providing common ground to facilitate communication.

The system lacked sufficient image quality and depth information which were cited as it's chief deficiencies. Origins of the low image quality were traced back to poor choices about the technology to be used made early in the design of the system. In a second iteration of the system I would rectify the shortcoming in the image quality by choosing a more suitable image compression method and hardware with greater graphics processing capabilities.

Further work has been planned to reorganise of the configuration of the camera and near-eye display on the helmet to make them independently adjustable and more comfortable and accessible to users. A comparative study of the HandsInAir system to it's counterpart the HandsOnVideo system has also been planned.

Although a two-dimensional workspace was satisfactory for communicating pointing gestures it was inadequate at clearly communicating more complex assembly instructions through the use of representational gestures. Recent advancements in depth sensing technology have made it feasible to explore the development of a three dimensional shared workspace that would enable participants greater freedom and range of expression. The use of depth sensing technology to implement more robust hand gesture recognition based on depth filtration instead of colour hue filtration will also be explored. The advanced detection mechanism would allow the helper to incorporate instructional apparatus into the shared workspace.

References

[1] Alem, L., Tecchia, F. and Huang, W. (2011) HandsOnVideo: Towards a gesture based mobile AR system for remote collaboration. In Alem, L. and Huang, W. (eds), Mobile collaborative augmented reality systems: Recent trends. Springer, NY, USA.

[2] Alem, L. and Li, J. (2011) A Study of Gestures in a Video-Mediated Collaborative Assembly Task. Advances in Human-Computer Interaction, vol. 2011, Article ID 987830, 7 pages.

[3] Alem, L., Hansen, S. And Li, J. (2006) Evaluating clinicians' experience in a telemedicine application: a presence perspective. Proceedings of the 18th Australia conference on Computer-Human Interaction, 47-54.

[4] Clark, H. H. and Brennan, S. E. (1991) Grounding in communication. In Perspectives on socially shared cognition. Washington, DC: American Psychological Association.

[5] Fussell, S. R., Setlock, L. D., Yang, J., Ou, J., Mauer, E. and Kramer, A. D. I. (2004) Gestures over video streams to support remote collaboration on physical tasks. Human-Computer Interaction, 19:273-309.

[6] Fussell, S. R., Setlock, L. D. and Parker, E. M. (2003) Where do helpers look? gaze targets during collaborative physical tasks. In Proceedings of extended abstracts on Human factors in computing systems (CHI'03), 768-769.

[7] Gergle, D., Kraut, R. E. and Fussell, S. R. (2006) The impact of delayed visual feedback on collaborative performance. In Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI '06), 1303-1312.

[8] Huang, W. and Alem, L. (2011) Supporting Hand Gestures in Mobile Remote Collaboration: A Usability Evaluation. Submitted to the 25th British Computer Society Conference on Human-Computer Interaction (BCS HCI2011).

[9] Kirk, D. S., Crabtree, A., and Rodden, T. (2005) Ways of the Hand. Proceedings of the ninth conference on European Conference on Computer Supported Cooperative Work (ECSCW'05), 1-21.

[10] Kirk, D. S., Rodden, T. and and Fraser, D. S. (2007) Turn it this way: grounding collaborative action with remote gestures. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07), 1039-1048.

[11] Kirk, D. S., and Stanton Fraser, D. (2005) The Impact of Remote Gesturing on Distance Instruction. Proceedings of the International Conference on Computer Supported Collaborative Learning, 301-310.

[12] Kirk, D. S., and Stanton Fraser, D. (2006) Comparing Remote Gesture Technologies for Supporting Collaborative Physical Tasks. In Proceedings of CHI Conference on Human Factors in Computing Systems, 1191-1200.

[13] Kraut, R. E., Fussell, S. R. and Siegel, J. (2003) Visual information as a conversational resource in collaborative physical tasks. Hum.- Comput. Interact., 18, 13-49.

[14] Kuzuoka, H.; Kosaka, J.; Yamazaki, K.; Suga, Y.; Yamazaki, A.; Luff, P. and Heath, C. (2004) Mediating dual ecologies. Proceedings of the 2004 ACM conference on Computer supported cooperative work, 477-486.

[15] Ou, J., Fussell, S. R., Chen, X., Setlock, L. D. and Yang, J. (2003) Gestural communication over video stream: supporting multimodal interaction for remote collaborative physical tasks. Proceedings of the 5th international conference on Multimodal interfaces, 242-249.

[16] Sakata, N.; Kurata, T.; Kato, T.; Kourogi, M. and Kuzuoka, H. (2003) WACL: supporting telecommunications using - wearable active camera with laser pointer. Proceedings of Seventh IEEE International Symposium on Wearable Computers, 53-56.

[17] Tang, A., Boyle, M. and Greenberg, S. (2004) Display and presence disparity in Mixed Presence Groupware. Proceedings of the fifth conference on Australian user interface, 73-82.

[18] (2011, Jun.). OpenCV [Online]. Available: http://opencv.willowgarage.com/wiki/

[19] (2011, Jun.). Independent JPEG Group [Online]. Available: http://www.ijg.org/

[20] (2011, Jun.). libjpeg-turbo [Online]. Available: http://libjpeg-turbo.virtualgl.org/

[21] (2011, Jun.). Windows Sockets 2 [Online]. Available: http://msdn.microsoft.com/en-us/library/ms740673(v=vs.85).aspx

[22] (2011, Jun.). Microsoft Foundation Classes [Online]. Available: http://msdn.microsoft.com/en-us/library/d06h2x6e(v=VS.100).aspx

[23] (2011, Jun.). Multithreaded Programming with the Event-based Asynchronous Pattern [Online]. Available: http://msdn.microsoft.com/en-us/library/hkasytyf.aspx

[24] (2011, Jun.). Critical Section Objects [Online]. Available: http://msdn.microsoft.com/en-us/library/ms682530(VS.85).aspx

[25] F. Dadgostar, A. Sarrafzadeh (2006) An adaptive real-time skin detector based on Hue thresholding: A comparison on two motion tracking methods. Pattern Recognition Letters
Volume 27, Issue 12, Pages 1342-1352

Appendix 1: Helper Questionnaire

Helper

Design and Evaluation of a Remote Guiding System

1) Indicate your preferred answer by (marking an "X" in the appropriate box of the seven-point scale. Please consider the entire scale when making your responses.

		Strongly disagree						Strongly agree
	Question	1	2	3	4	5	6	7
1	I found the system easy to learn							
2	I found the system easy to use							
3	I found the system intuitive to use							
4	I found the system enjoyable to use							
5	I think the system is useful for remote guiding tasks							
6	I was satisfied with my task performance							
7	I found easy to communicate with my partner (audio and video)							
8	I felt that my partner and I were at the same location							
9	I was satisfied with my interaction with my partner							
10	I found it easy to point to objects using my hands							
11	I found it easy to demonstrate assembly of objects using my hands							
12	I found it easy to guide my partner to look around his workspace							
13	I found it is easy to guide when I faced to the wall							
14	I found it is easy to guide when I sat at a desk							

2) What are pros and cons of using the system to remotely guide your partner to perform the object assembly task? Please give details.

3) What are your opinions about your experience using your hands to guide your partner? Please give details.

4) In what ways do you think the system can be improved? Please give details.

5) Do you have any additional comments?

Appendix 2: Worker Questionnaire

Worker (Rugged system)

Design and Evaluation of a Remote Guiding System

1) Indicate your preferred answer by (marking an "X" in the appropriate box of the seven-point scale. Please consider the entire scale when making your responses.

		Strongly disagree						Strongly agree
	Question	1	2	3	4	5	6	7
1	I found the system easy to learn							
2	I found I the system easy to use							
3	I found the system intuitive to use							
4	I found the system enjoyable to use							
5	I think the system is useful for remote guiding tasks							
6	I am satisfied with my task performance							
7	I found it easy to communicate with my partner (audio and video)							
8	I felt that my partner and I were at the same location							
9	I was satisfied with my interaction with my partner							
10	I felt aware of my physical surroundings while using the system							
11	I felt confidant avoiding obstacles while walking around							
12	I found it easy to understand which object my partner is pointing to							
13	I found easy to follow my partner’s hand gestures to assembly objects							
14	Wearing the system was comfortable for the duration of the task							

2) What are pros and cons of using the system to be guided in performing the object assembly task? Please give details.

3) What are your opinions on seeing the hands of the person helping you?

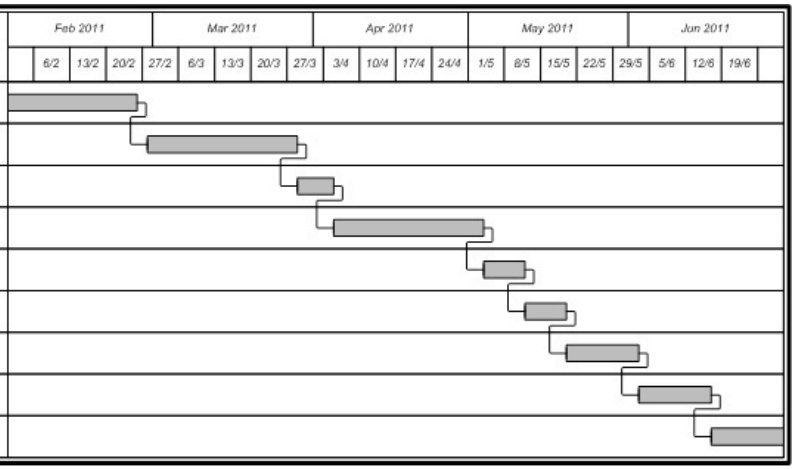
4) What was your experience using the near-eye display for instructions while communicating with your partner on the task? Please give details.

5) How do you think the system can be improved? Please give details.

6) Do you have any other comments?

Intended Work Schedule

ID	Task Name	Start	Finish	Duration
1	Design and Analysis - HandsInAir	2/1/2011	2/25/2011	19d
2	Implementation - HandsInAir	2/28/2011	3/28/2011	21d
3	Testing Within Team	3/29/2011	4/4/2011	5d
4	Design and Implementation – HandsInAir	4/5/2011	5/3/2011	21d
5	Design Formal Evaluation	5/4/2011	5/11/2011	6d
6	Conduct Formal Evaluation	5/12/2011	5/19/2011	6d
7	Explore integration with 3D mapping	5/20/2011	6/2/2011	10d
8	Implement 3D integration	6/3/2011	6/16/2011	10d
9	Explore further extension to HandsInAir	6/17/2011	6/30/2011	10d



Actual Work Schedule

ID	Task Name	Start	Finish	Duration
1	Design and Analysis - HandsInAir	2/1/2011	2/15/2011	11d
2	Implementation - HandsInAir	2/16/2011	3/23/2011	26d
3	Testing Within Team	3/24/2011	4/5/2011	9d
4	Design and Implementation – HandsInAir	4/6/2011	5/31/2011	40d
5	Design Formal Evaluation	6/1/2011	6/16/2011	12d
6	Conduct Formal Evaluation	6/17/2011	6/21/2011	3d
7	Explore integration with 3D mapping	6/22/2011	7/1/2011	8d
8	Implement 3D integration	7/4/2011	7/7/2011	4d