ABSTRACT

Title of dissertation: AUGMENTED REALITY
FOR SPACE APPLICATIONS
Massimiliano Di Capua, Master of Science, 2008

Dissertation directed by: Professor David L. Akin
Department of Aerospace Engineering
Space Systems Laboratory

Future space exploration will inevitably require astronauts to have a higher degree of autonomy in decision-making and contingency identification and resolution. Space robotics will eventually become a major aspect of this new challenge, therefore the ability to access digital information will become crucial for mission success. In order to give suited astronauts the ability to operate robots and access all necessary information for nominal operations and contingencies, this thesis proposes the introduction of In-Field-Of-View Head Mounted Display Systems in current Extravehicular Activity Spacesuits. The system will be capable of feeding task specific information on request, and through Augmented Reality technology, recognize and overlay information on the real world for error checking and status purposes. The system will increase the astronaut's overall situational awareness and nominal task accuracy, reducing execution time and human error risk. The aim of this system is to relieve astronauts of trivial cognitive workload, by guiding and checking on them in their operations. Secondary objectives of the system will be the introduction of electronic checklists, and the ability to display the status of the suit

and surrounding systems as well as interaction capabilities. Features which could be introduced are endless due the nature of the system, allowing extreme flexibility and future evolution without major design changes. This work will focus on the preliminary design of an experimental Head Mounted Display and its testing for initial evaluation and comparison with existing information feed methods. The system will also be integrated and tested in the University of Maryland Space Systems Laboratory MX-2 experimental spacesuit analogue.

AUGMENTED REALITY FOR SPACE APPLICATIONS

by

Massimiliano Di Capua

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2008

Advisory Committee:
Professor David L. Akin, Chair/Advisor
Professor Raymond J. Sedwick
Professor Derek A. Paley

Acknowledgments

My most sinceere gratitude goes to Dr. Akin for allowing me to embark in this adventure and for guiding me through the process. Many thanks go to all the SSLers who make the Space Systems Laboratory a wonderfull place to work. Special thanks go to all the people who participated in the experiment and who volunteered to dedicate their time for this study. In particular my gratitude goes to Sharon Singer for all the precious feedback and for taking the time in reading and editing my work. I also wish to thank Martin Stolen, for all his patience, mentoring and all the hours spend listening to me. Finally to all the MX-2 team for giving me the inspiration and making me feel part of the team. Many thanks also go to Dr. Bowden and Rosalia K. Webb for their administrative support. Without their help all this would have never happened. Finally, to my family for their unconditional love and trust. Although far they were always there when i most needed them.

Table of Contents

List of Figures

List of Abbreviations

AR	Augmented Reality
CCA	Communications Carrier Assembly
CS	Control Station
DCM	Display and Control Module
DMU	Data Management Unit
EMU	Extravehicular Mobility Unit
EVA	Extra Vehicular Activity
HMD	Head Mounted Display
HUD	Head Up Display
HUT	Hard Upper Torso
IPD	Interpupillary Distance
IRB	Institutional Review Board
ISS	International Space Station
IMU	Inertial Measurment Unit
IVA	Intravehicular Activity
LCD	Liquid Crystals Display
MCC	Mission Control Center
MR	Mixed Reality
NASA	National Aeronautics & Space Administration
RCL	RANGER Communications Layer
SSL	Space Systems Laboratory
UMD	University of Maryland
VE	Virtual Environments
VR	Virtual Reality

Chapter 1
INTRODUCTION

Space exploration will be this century's great adventure. Hopefully it will see people walking once again on the Moon, perhaps living there, establishing outposts, and roving the lunar surface in search for answers to the many scientific questions that fill our minds. Perhaps it will also see people on Mars, in search of traces of life. Many are the quests that await ahead, all with a common trace: human presence. Scientists and engineers are today involved in defining these missions, and addressing the technical difficulties and goals; in brief, setting the baseline for tomorrow's philosophy of space exploration.

1.1 Problem Statement

This thesis wishes to contribute to this effort by studying a visual information display system that would increase astronaut's overall situational awareness, grant them a higher level of independence from mission control, and enable a more intuitive and efficient human-robot interface. The above are just a small set of the possibilities of what the introduction of an in-field-of-view digital display system could deliver. The main reason for devoting attention to this subject lies in the shortcomings of current mission planning and execution methodology and their requirements, which will probably be inapplicable on a Lunar or Martian mission. The current mission planning for EVA operations relies on two main aspects:

1.1.1 Training :

Astronauts are trained thoroughly in every detail of the mission they are going to fly. They choreograph and rehearse the mission for several months before launch. Training focuses mainly on nominal tasks, and problem diagnosis and resolution. The knowledge acquired during this preparation phase will serve as a baseline for the mission. Although this approach has been very successful in the past, it has limitations that are intrinsic in human nature. While the reliability of astronaut's memory decreases as the information load increases, the affordable training duration cannot increase without limit. Also the number of contingency scenarios, and therefore mission complexity, are related to the mission duration and number of systems to be handled, making these new missions potentially more complex. Also intrinsic in space flight is the reality that not all contingency scenarios can be predicted. Training provides the astronauts the necessary skills and knowledge to attempt the diagnosis and resolution of possible failures, and when this fails, astronauts currently rely on mission control for procedures and instructions. In brief, if the current training philosophy were to be applied in preparing for future planetary missions, it would inevitably lead to a more generic training, leaving to the astronauts the responsibility of making decisions on their own, possibly without the necessary information, since communications might be prohibitively delayed.

1.1.2 Ground Communications :

Radio communications between astronauts and mission control, play an important role in nominal scenarios for purposes of control and verification, but these communications become crucial during contingencies, when they provide semi-realtime aid to the astronauts. This means of assistance has been extensively used in the past, and it is still the main (and almost only) external source of information for the astronauts. Unfortunately, as missions move further from Earth, time delays between astronauts and mission control will increase until the point where radio communications will no longer be acceptable for this purpose. This could leave the astronauts on their own for several minutes therefore they must be given the means to indipendently acquire the necessary information to diagnose and resolve contingencies rapidly.

1.2 Information Feed Methods :

During the Apollo missions, astronauts were equipped with a small book containing checklists and task-specific information. This system revealed itself to be very useful, although not optimal by alowing the astonauts to acess critical information withouth relying on just their memory. Astronauts had to handle this booklet from within their spacesuits, but mobility was restricted and searching for information was difficult and time consuming. Also of note, the information provided by this means was limited, and required the use of at least one hand. Astronauts still use this method because it's simple and reliable, although they prefer to talk to mission control for help. A full library is needed on current spacecraft to provide astronauts with all the relevant information, but Volume and Mass constraints prohibit this from being available in book form to an astronaut in both EVA and IVA. Expansion and flexibility of this system is also limited making it not the ideal choice for future long duration missions. Despite all the drawbacks of this method, it still finds application as a back-up system; therefore, the proposed systems are not aimed at replacing the previous methods, but merely optimizing and expanding them.

Today, we hardly see analogies between space flight and everyday life, but interestingly, solutions to the problems above can be found in our daily behaviors. In the last fifty years, we have seen the rapid evolution of computers and later networks, which changed our approach to information research and availability as possibilities have expanded. Today, we cannot imagine what it would be like to conduct our everyday lives without a computer and the internet. Computers and wireless technology enable us to access almost all of human knowledge from anywhere in the globe, greatly enhancing our autonomy. Why not introduce the same concepts into future space suits? First of all there are many technological challenges that need to be addressed in order to allow us to introduce theese concepts in the space suit environment. Another major difficulty lies in the the interfaces that can be used. Electronics difficulties are intrinsic in the development of any space-rated system, often the result of stringent limitations on dimensions and masses. Mobility is also highly restricted, therefore interfaces such as keyboards and mice are inapplicable, leading to the necessity of new solutions.

This work attempts to define the capabilities that an enabling system should have, as well as define possible means to implement it using current off-the-shelf technology and components. It will also attempt to define the strengths and weaknesses of two proposed systems, as well as hopefully justify the introduction of a higher complexity system as compared to traditional methods of conveying information, such as booklets. This study will eventually set the baseline for a more detailed and focused development of future in-suit digital information displays and implementable features, such as augmented reality.

1.3 Proposed Systems Overview

1.3.1 *Head-Up Display (HUD):*

This system was the first to be developed, and it relies on an off-the-shelf modified LCD screen (Specifications can be founf in Appendix A). Its purpose is to deliver information to suited astronauts without obstructing their primary field of view. The system is connected to a computer that manages the displayed content and through which the astronaut interacts via voice commands. This system was designed specifically for integration in the MX-2 spacesuit analogue for validation and testing.

1.3.2 *Head-Mounted Display (HMD):*

This second system is an evolution of the HUD which allows the suited astronaut to access information within their primary field-of-view with limited obstruction. The system is also designed to deliver all the HUD features, as well as to recognize what the astronaut is looking at and overlay synthetic content on objects. The system is composed of several subsystems such as a head-mounted visor, a webcam, and a head mount ("snoopy-cap").

1.4 Thesis Structure

This thesis is divided into five chapters. The next chapter will focus on describing the previous work done in the matter of space suit information feed methods, as well as augmented reality techniques. Chapter Three will describe the systems design phase by initially defining and describing the requirements derived from the previous chapter, and how they were met. Chapter Four will cover the testing of the proposed systems, describing the methods and procedures, as well as the results achieved. Finally, chapter Five will include my conclusions and proposed future research in the matter. Appendices are provided, including code, data-sheets and documentation.

Chapter 2
BACKGROUND AND PREVIOUS WORK
2.1 Augmented Reality

Augmented Reality (AR) is a variation of Virtual Environments (VE), or Virtual Reality (VR) as it is more commonly called. VE technologies completely immerse the user inside a synthetic environment. While immersed, the user cannot see the real world around him. In contrast, AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world. Therefore, AR supplements reality, rather than completely replacing it.

Figure 2.1: Paul Milgram and Fumio Kishino: Virtuality Continuum

Figure 2.1 represents a concept in computer science that there is a continuous scale ranging between the completely virtual,and the completely real. The reality-virtuality continuum encompasses all possible variations and compositions of real and virtual objects. The concept was first introduced by Paul Milgram. The area between the two extremes, where both the real and the virtual are mixed, is the so-called Mixed or Augmented reality. While immersed in an AR environment, it would appear to the user that the virtual and real objects coexisted in the same space. Figure 2.2 shows an example of what this might look like. The AR environment includes real elements such as the astronaut, the rover and the background as well as synthetic elements such as the 3-D map, suit and system status and video feed from the rover camera. Although in this work AR will be referred to exclusively as visual augmentation, it is important to understand that this concept can be extended to all the human senses [1].

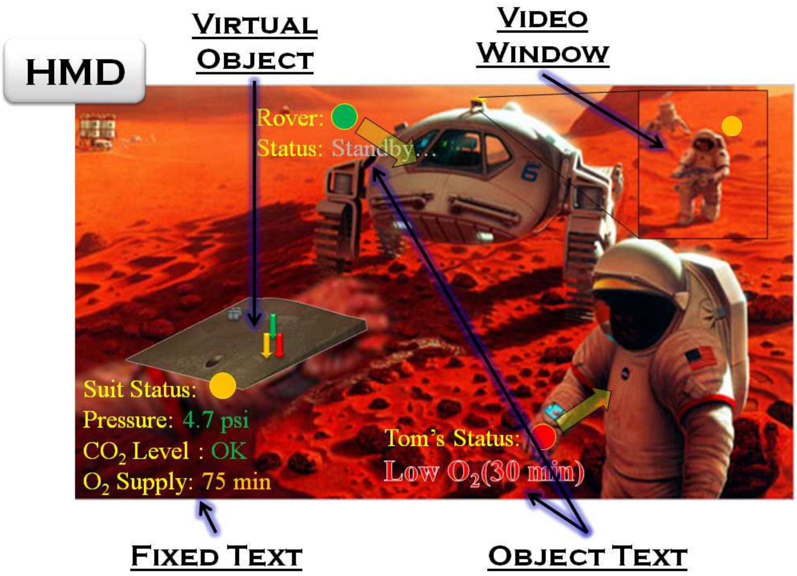


Figure 2.2: Martian EVA concept with AR overlays. (courtesy of NASA\Pat Rawlings, SAIC)

2.1.1 Definition

Some define AR in a way that requires the use of Head-Mounted Displays (HMDs). To avoid limiting AR to specific technologies,AR systems are here defined to have the following three characteristics:

- Combines real and virtual objects in a real environment;
- Runs interactively, and in real time;
- Registers (aligns) real and virtual objects with each other in the 3-D space.

This definition includes other technologies other than see-through HMDs such as monitor–based interfaces, monocular systems, and various other combining technologies while retaining the essential components of AR [2].

2.1.2 Augmented Reality Space Related Applications

2.1.2.1 Life Support Control and Comfort Control

Life support control and comfort control for the EMU are provided by the chest-mounted DCM, which can interfere with the work area of the astronaut. Life support information can be viewed by the EVA astronaut on the DCM or by support personnel monitoring telemetered lifesupport data. Current limitations of the DCM display make its presentation of life support data useful primarily for intermittent status checks in which it is scrolled through the various parameters of interest or for the investigation of alarms or warnings automatically generated by fault detection logic. Routine awareness of activity levels, thermal state, expendables status, etc., is primarily maintained by support personnel and, to the extent that the EVA astronaut's involvement is required, communicated by voice link. Future EVA information interfaces for life support control and comfort control are likely to maintain dedicated controls for critical life support functions. Integration of life support status information into improved information displays could reduce communications chatter, and provide the capability for more autonomous operation in environments with reduced real-time mission-control support (e.g., reduced local mission support crewmembers and/or long time delays). This would provide astronauts with the capability to monitor and manage their own work rates and to manage constraints including thermal loads and consumables margins [13].

2.1.2.2 Mission and Task Planning

During current operations, astronauts participate in mission and task planning as part of their training, but do not generally plan whole sequences of actions or tasks on-orbit because of the extensive verification and validation required to develop a viable EVA plan. Current operations can be described as an attempt to dance a ballet. The goal of the EVA is to execute a predetermined, practiced plan, follow rehearsed contingencies when necessary, and improvise only as absolutely required. Future EVAs are likely to entail dramatically increased task uncertainty and an increased number of potential contingencies, making current levels of preparation impractical. EVA astronauts will take on a larger mission and task planning role by making observations and measurements that will affect the remainder of their EVA goals and objectives. One example might be the inspection of a failed piece of equipment external to the ISS; the remaining goals and objectives for the EVA could depend strongly on a surface inspection or electrical measurements made by the astronaut. During planetary exploration, a geologic traverse could be driven largely by observations made earlier in the geologic traverse. Thus, information interfaces should enable astronauts to acquire, record, analyze, and communicate the data required to support mission and task planning. In addition, when more autonomous operations are required, EVA information interfaces should enable astronauts to perform their own mission and task planning to the extent required to maximize the value of EVA within operational constraints. With one-way line-of-sight light-travel-time delays of 4.5 to 21 minutes between Earth and Mars, EVA operations on the Martian surface are likely to be autonomous or semi-autonomous. One might imagine a geologist replanning the rest of their geologic traverse for the day based on an important discovery. The geologist might use an information interface to specify limited temporal, spatial, and other characteristics of their new planned traverse and submit an EVA plan (analogous to a pilot filing a flight plan) after validating that the new traverse meets all applicable flight rules including expected thermal loads and consumables margins [13].

2.1.2.3 Localization and Situational Awareness

Localization and situational awareness continue to be a problem in microgravity EVA operations due to spatial disorientation (including inversion illusions), a lack of direction cues, contrast challenges, and limited visibility (especially to the rear of the spacesuit). While the same orientation challenges experienced in microgravity are not encountered in partial gravity, the Apollo lunar surface astronauts had difficulty with localization because of the undulating and self-similar nature of the lunar surface. The EVA information interface should assist in localization by providing cues to astronaut orientation and position relative to visible landmarks. Likewise, information interfaces should enhance situational awareness by making available to the astronaut basic status information (for example, time on EVA or time on task), progress compared to plan, consumables, upcoming events (for example, time to events like sun-up or sun-down), or other contextual information. Visual displays could also highlight landmarks, keep-out zones, or other hazards, and could illustrate the location and characteristics of other EVA events. Providing behind-the-back clearance sensing might also be useful to prevent astronauts from bumping into and potentially damaging objects that are behind them. Many traditional localization techniques can be applied to planetary surface operations, including identification of landmarks, observation of sun-angle, radiolocation, and localization schemes like the global positioning system (GPS). Because localization is likely to be such a routine activity during planetary surface exploration, EVA information interfaces for localization should be highly automated. Localization schemes can be built upon surface-based communications and networking infrastructure if a GPS-like localization scheme is not available. Information interfaces for planetary exploration could use a traditional map view to illustrate topography, landmarks, keep-out zones, locations of ongoing EVA events, and temporal events such as comparing actual progress on a geologic traverse to the nominal planned traverse. A visual display might serve as the EVA equivalent for planetary exploration to today's multi-function flight displays for pilots, integrating temporal and spatial data into a single view [13].

2.1.2.4 Navigation

During microgravity operations, translation routes are learned during ground-based training or via study prior to EVA. Nevertheless, translation can be disorienting over significant distances on large space structures. During Apollo lunar surface exploration, navigation was hampered by a lack of landmarks, self-similarity of the terrain, reduced line-of-sight distances, and visual challenges with some sun-relative directions of travel [14] [15]. In addition, reliance on dead-reckoning navigation sharply reduced navigational accuracy until the deployment of the lunar rover. During microgravity operations, visual display of preferred translation routes or techniques could be used to assist astronauts while translating from site to site, for example, during an unplanned EVA. Extensive surface data is likely to be available for most, if not all, future planetary surface missions. An integrated display of these data combined with aforementioned data such as landmarks, hazards, and keep-out zones could provide a highly functional aid to navigation [13].

2.1.2.5 Task Execution

Task execution during extravehicular activity has been enhanced by continued (but limited) improvement in mobility of spacesuit joints and gloves, standardization of mechanical interfaces, and evolution of a limited but powerful set of tools that provide position and orientation control and mechanical advantage. Task execution often includes one or more (and often many repetitions of) steps including physical manipulation, measurement, recording, processing, communication and verification. During Apollo, astronauts read out measurements from a gravimeter over their radios. LEO operations still utilize the same techniques when tightening a bolt, an EVA astronaut will count out loud the number of cranks and degrees per crank made while turning a torque wrench. EVA information interfaces should be developed that reduce the time and energy (mental and physical) required to execute a task. Significantly improved data automation is required to achieve this goal by improving the task efficiency or the efficiency with which task outcomes are communicated. Efficient task execution requires that EVA astronauts have access to accurate task-related information, especially for complex tasks or for tasks for which an astronaut has not recently trained. EVA information interfaces could deliver video, text, and graphics, possibly acquired in real time over a wireless network, to the astronauts. These information interfaces should also permit real-time collaboration among members of the EVA team to support routine discussions, troubleshooting, and contingency or emergency operations. Delivery or display of information could be initiated by a remote operator on request by an EVA astronaut [7] in response to voice commands, [9], or based on contextual data such as tool usage, position, orientation, posture, or time. For example, an electronic torque wrench could measure and wirelessly transmit the total number of degrees of rotation it has been turned since being reset. Grasping or resetting the torque wrench could wirelessly activate a torque-wrench display, and data from the device could be displayed on a visual display in the space suit or viewed by support personnel. In the context of planetary exploration, geographical information systems (GIS) may serve as a useful model for automation of many of the components of task execution. Physical manipulations, observations and measurements can be tied to contextual data (such as position, time, etc.) and integrated into a virtual world model that can be subsequently analyzed or communicated. Research should be devoted to enable activities that are common during terrestrial field work such as imaging, note taking and sketching. Interfaces for these activities may require some physical movement or may be based on voice commands [13].

2.1.2.6 EVA Human-Robot Interfaces:

The future EVA astronaut will not be without helping hands. Robotic systems are currently used mainly to position astronauts in manned human-robotic on-orbit operations, but will likely fulfill a multitude of functions in future EVA activity. In fact recent work indicates that robots can successfully be employed together with their human counterparts in complex operations like repairing the Hubble Space Telescope [16]. The robots would be teleoperated from Earth, and perform simple routine tasks like site preparation and cleanup, but also more dexterous tasks like fastening bolts and supplying tools to the EVA astronaut. Teleoperated robots are also being equipped with dexterous hands and immersive teleoperation interfaces, that give them capabilities similar to those of EVA astronauts [17].

Some work has also been performed on allowing the EVA astronaut to take direct control of robotic systems, including ESA's EVA Man Machine Interface (EMMI). This is a portable teleoperation interface designed for the European Robotic Arm (ERA) aimed for use at the International Space Station (ISS). More work is needed to assess the most suitable type of input device for teleoperation from an EVA suit however.

A similar robotic presence is envisioned for potential lunar exploration. To expand its capabilities, increase its safety, and augment its operations, the lunar outpost will by necessity have to incorporate extensive use of robotic systems. Given the short speed-of-light time delay from the Moon to the Earth (usually around a minute), it makes little sense to have humans in a local habitat directing robots when Earth-based support crew can perform most robotic control tasks just as well. Getting human eyes, hands, and minds to the exploration or development sites will involve learning to work efficiently and effectively in planetary surface EVA, including direct interactions with supporting robotic systems. Certainly, the primary operating mode of these robots will be autonomous, directed and monitored by the local EVA crew. As tasks become more difficult or the robots encounter unplanned conditions, the human/robot interaction will have to move from high-level supervisory control (e.g., come here) to lower-level command structures (e.g., go to the right of that rock, then back to the left) to full teleoperation (e.g., remote driving). The most effective human to issue these commands is the human standing on the site; a human in a pressure suit. They must receive disparate data in multiple forms, cognitively process it while incorporating a knowledge base of objectives, procedures, and diagnostics, and issue commands ranging from high-level goal direction to specific motions of individual actuators. Thus, in future space operations spacesuits will become portable command and control stations for the accompanying robotic systems [12].

2.1.3 Augmented Reality Interfaces

A basic design decision in building an AR system is how to accomplish the combining of real and virtual. Two basic choices are available: optical and video technologies. Each has particular advantages and disadvantages. This section compares the two and notes the tradeoffs.

2.1.3.1 Video Interfaces

Video see-through HMDs work by combining a closed-view HMD with one or two head-mounted video cameras. The video cameras provide the user's view of the real world. Video from these cameras is combined with the graphic images created by the scene generator, blending the real and virtual. The result is sent to the monitors in front of the user's eyes in the closed-view HMD. Figure 2.3 shows a conceptual diagram of a video see-through HMD.

Video composition can be done in more than one way. A simple way is to use chroma-keying, a technique used in many video special effects. The background of the computer graphic images is set to a specific color, say green, which none of the virtual objects use. Then the combining step replaces all green areas with the corresponding parts from the video of the real world. This has the effect of superimposing the virtual objects over the real world. A more sophisticated composition would use depth information. If the system had depth information at each pixel for the real world images, it could combine the real and virtual images by a pixel-by-pixel depth comparison. This would allow real objects to cover virtual objects and vice-versa. AR systems can also be built using monitor-based configurations, instead of HMDs. Figure 2.4 shows how a monitor-based system might be built.

In this case, one or two video cameras view the environment. The cameras may be static or mobile. In the mobile case, cameras mounted to a robot would move, with their locations tracked. The video of the real world and the graphic images generated by a scene generator are combined, just as in the video see-through HMD

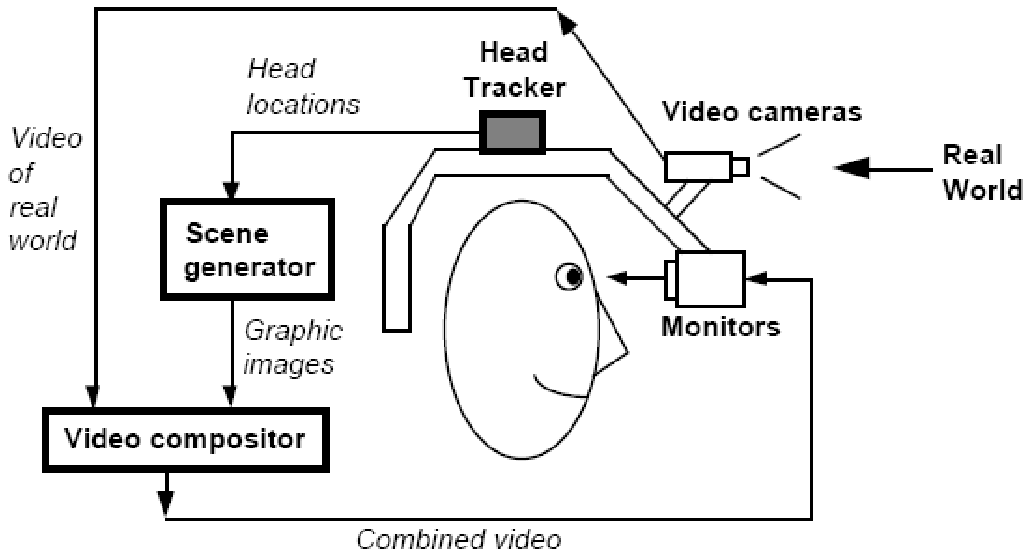


Figure 2.3: Video interface conceptual diagram. [1]

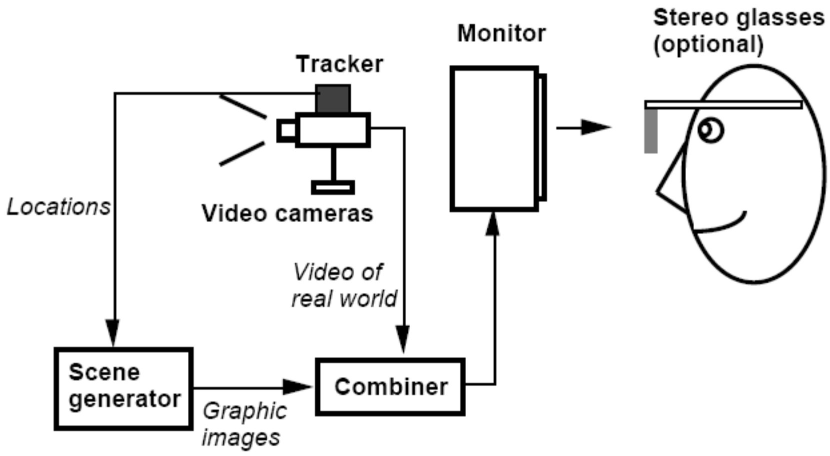


Figure 2.4: Monitor-Based video interface conceptual diagram. [1]

case, and displayed in a monitor in front of the user. The user does not wear the display device. Optionally, the images may be displayed in stereo on the monitor, which then requires the user to wear a pair of stereo glasses.

2.1.3.2 Optical Interfaces

Optical see-through HMDs work by placing optical combiners in front of the user's eyes. These combiners are partially transmissive, so that the user can look directly through them to see the real world. The combiners are also partially reflective, so that the user sees virtual images bounced off the combiners from head-mounted monitors. This approach is similar in nature to Head-Up Displays (HUDs) commonly used in military aircraft, except that the combiners are attached to the head. Figure 2.5 shows a conceptual diagram of an optical see-through HMD.

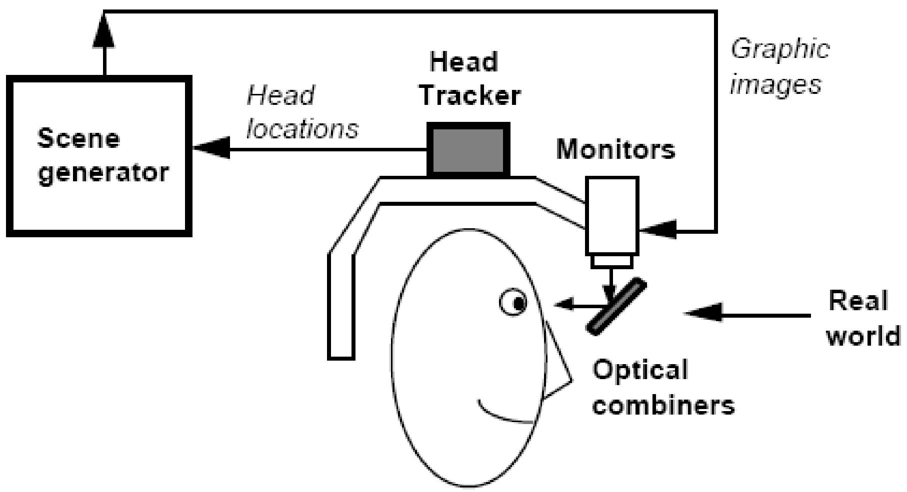


Figure 2.5: Optical see-through interface conceptual diagram. [1]

The optical combiners usually reduce the amount of light that the user sees from the real world. Since the combiners act like half-silvered mirrors, they only let in some of the light from the real world, so that they can reflect light from the monitors into the user's eyes. For example, the HMD used later in this study, transmits about 30 percent of the incoming light from the real world. The level of blending is a design parameter. More sophisticated combiners might vary the level of contributions based upon the wavelength of light. For example, such a combiner might be set to reflect all light of a certain wavelength and none at any other wavelengths. This would be ideal with a monochrome monitor. Virtually all the light from the monitor would be reflected into the user's eyes, while almost all the light from the real world (except at the particular wavelength) would reach the user's eyes. However, most existing optical see-through HMDs do reduce the amount of light from the real world, so they act like a pair of sunglasses when the power is cut off.

2.1.3.3 Optical/Video Interface Comparison

Both optical and video technologies have their roles, and the choice of technology depends on the application requirements. An optical approach has the following advantages over a video approach:

1. Simplicity: Optical blending is simpler and cheaper than video blending. Optical approaches have only one stream of video to worry about: the graphic images. The real world is seen directly through the combiners, and that time delay is generally a few nanoseconds. Video blending, on the other hand, must deal with separate video streams for the real and virtual images. Both streams have inherent delays in the tens of milliseconds. Digitizing video images usually adds at least one frame time of delay to the video stream, where a frame time is how long it takes to completely update an image. A monitor that completely refreshes the screen at 60 Hz has a frame time of 16.67 ms. The two streams of real and virtual images must be properly synchronized or temporal distortion results. Also, optical see-through HMDs with narrow field-of-view combiners offer views of the real world that have little distortion. Video cameras almost always have some amount of distortion that must be compensated for, along with any distortion from the optics in front of the display devices. Since video requires cameras and combiners that optical approaches do not need, video will probably be more expensive and complicated to build than optical-based systems.

2. Resolution: Video blending limits the resolution of what the user sees, both real and virtual, to the resolution of the display devices. Optical see-through also shows the graphic images at the resolution of the display device, but the user's view of the real world is not degraded. Thus, video reduces the resolution of the real world, while optical see-through does not.

3. Safety: Video see-through HMDs are essentially modified closed-view HMDs. If the power is cut off, the user is effectively blind. This is a safety concern

in some applications. In contrast, when power is removed from an optical see-through HMD, the user still has a direct view of the real world. The HMD then becomes a pair of heavy sunglasses, but the user can still see.

4. No eye offset: With video see-through, the user's view of the real world is provided by the video cameras. In essence, this puts his "eyes" where the video cameras are. In most configurations, the cameras are not located exactly where the user's eyes are, creating an offset between the cameras and the real eyes. The distance separating the cameras may also not be exactly the same as the user's interpupillary distance (IPD). This difference between camera locations and eye locations introduces displacements from what the user sees compared to what he expects to see. For example, if the cameras are above the user's eyes, he will see the world from a vantage point slightly taller than he is used to. Video see-through can avoid the eye offset problem through the use of mirrors to create another set of optical paths that mimic the paths directly into the user's eyes. Using those paths, the cameras will see what the user's eyes would normally see without the HMD. However, this adds complexity to the HMD design.

   Offset is generally not a difficult design problem for optical see-through displays. While the user's eye can rotate with respect to the position of the HMD, the resulting errors are negligible. Using the eye's center of rotation as the viewpoint in the computer graphics model should eliminate any need for eye tracking in an optical see-through HMD.

Video blending offers the following advantages over optical blending:

1. Flexibility in composition strategies: A basic problem with optical see-through is that the virtual objects do not completely obscure the real world objects, because the optical combiners allow light from both virtual and real sources. Building an optical see-through HMD that can selectively shut out the light from the real world is difficult. In a normal optical system, the objects are designed to be in focus at only one point in the optical path: the user's eye. Any filter that would selectively block out light must be placed in the optical path at a point where the image is in focus, which obviously cannot be the user's eye. Therefore, the optical system must have two places where the image is in focus: at the user's eye and the point of the hypothetical filter. This makes

the optical design much more difficult and complex. No existing optical see-through HMD blocks incoming light in this fashion. Thus, the virtual objects appear ghost-like and semi-transparent. This damages the illusion of reality because occlusion is one of the strongest depth cues.

In contrast, video see-through is far more flexible about how it merges the real and virtual images. Since both the real and virtual are available in digital form, video see-through compositors can, on a pixel-by-pixel basis, take the real, or the virtual, or some blend between the two to simulate transparency. Because of this flexibility, video see-through may ultimately produce more compelling environments than optical see-through approaches.

2. Wide field-of-view: Distortions in optical systems are a function of the radial distance away from the optical axis. The farther one looks away from the center of the view, the larger the distortions get. A digitized image taken through a distorted optical system can be undistorted by applying image processing techniques to unwarp the image, provided that the optical distortion is well characterized. This requires significant amounts of computation. It is harder to build wide field-of-view displays with optical see-through techniques. Any distortions of the user's view of the real world must be corrected optically, rather than digitally, because the system has no digitized image of the real world to manipulate. Complex optics are expensive and add weight to the HMD. Wide field-of-view systems are an exception to the general trend of optical approaches being simpler and cheaper than video approaches.

3. Real and virtual view delays can be matched: Video offers an approach for reducing or avoiding problems caused by temporal mismatches between the real and virtual images. Optical see-through HMDs offer an almost instantaneous view of the real world but a delayed view of the virtual. This temporal mismatch can cause problems. With video approaches, it is possible to delay the video of the real world to match the delay from the virtual image stream.

4. Additional registration strategies: In optical see-through, the only information the system has about the user's head location comes from the head tracker. Video blending provides another source of information: the digitized image of the real scene. This digitized image means that video approaches can employ additional registration strategies unavailable to optical approaches.

5. Easier to match the brightness of real and virtual objects.

2.1.3.4 Image Focus and Contrast

Image focus can be a problem for both optical and video approaches. Ideally, the virtual should match the real. In a video-based system, the combined virtual and real image will be projected at the same distance by the monitor or HMD optics. However, depending on the video camera's depth-of-field and focus settings, parts of the real world may not be in focus. In typical graphics software, everything is rendered with a pinhole model, so all the graphic objects, regardless of distance, are in focus. To overcome this, the graphics could be rendered to simulate a limited depth-of-field, and the video camera might have an autofocus lens. In the optical case, the virtual image is projected at some distance away from the user. This distance may be adjustable, although it is often fixed. Therefore, while the real objects are at varying distances from the user, the virtual objects are all projected to the same distance. If the virtual and real distances are not matched for the particular objects that the user is looking at, it may not be possible to clearly view both simultaneously.

Contrast is another issue because of the large dynamic range in real environments and in what the human eye can detect. Ideally, the brightness of the real and virtual objects should be appropriately matched. Unfortunately, in the worst case scenario, this means the system must match a very large range of brightness levels. The eye is a logarithmic detector, where the brightest light that it can handle is about eleven orders of magnitude greater than the smallest, including both dark-adapted and light-adapted eyes. In any one adaptation state, the eye can cover about six orders of magnitude. Most display devices cannot come close to this level of contrast. This is a particular problem with optical technologies, because the user has a direct view of the real world. If the real environment is too bright, it will wash out the virtual image. If the real environment is too dark, the virtual image will wash out the real world. Contrast problems are not as severe with video, because the video cameras themselves have limited dynamic response, and the view of both the real and virtual is generated by the monitor, so everything must be clipped or compressed into the monitor's dynamic range.

2.1.3.5 Portability

In almost all Virtual Environment systems, the user is not encouraged to walk around much. Instead, the user navigates by "flying" through the environment, walking on a treadmill, or driving some mockup of a vehicle. Whatever the technology, the result is that the user stays in one place in the real world. On the contrary space related AR applications, will need to support a user who will move in a large environment. In space related applications, usually AR scenarios require the user to be at the place where the task is to take place, Robot teleoperations are an exception, but will be regarded as a simplified case of nominal EVA AR applications. Therefore, AR systems will place a premium on portability. The scene generator, the HMD, and the tracking system must all be self-contained and capable of surviving exposure to the environment.

2.1.3.6 Registration

One of the most basic problems currently limiting Augmented Reality applications is the registration problem. The objects in the real and virtual worlds must be properly aligned with respect to each other, or the illusion that the two worlds coexist will be compromised. Registration problems also exist in Virtual Environments, but they are not nearly as serious because they are harder to detect than in Augmented Reality. Since the user only sees virtual objects in VE applications, registration errors result in visual-kinesthetic and visual-proprioceptive conflicts. Such conflicts between different human senses may be a source of motion sickness. Because the kinesthetic and proprioceptive systems are much less sensitive than the visual system, visual-kinesthetic and visual-proprioceptive conflicts are less noticeable than visual-visual conflicts. For example, a user wearing a closed-view HMD might hold up her real hand and see a virtual hand. This virtual hand should be displayed exactly where she would see her real hand, if she were not wearing an HMD. But if the virtual hand is wrong by five millimeters, she may not detect that unless actively looking for such errors. The same error is much more obvious in a see-through HMD, where the conflict is visual-visual. Furthermore, a phenomenon known as visual capture makes it even more difficult to detect such registration errors. Visual capture is the tendency of the brain to believe what it sees rather than what it feels, hears, etc. That is, visual information tends to override all other senses.

Registration errors are difficult to adequately control because of the high accuracy requirements and the numerous sources of error. These sources of error can be divided into two types: static and dynamic. Static errors are the ones that cause registration errors even when the user's viewpoint and the objects in the environment remain completely still. Dynamic errors are the ones that have no effect until either the viewpoint or the objects begin moving. For current HMD-based systems, dynamic errors are by far the largest contributors to registration errors, but static errors cannot be ignored either.

2.2 Vision Based Sensing Techniques

Registration based solely on the information from inertial tracking systems is like building an "open-loop" controller. The system has no feedback on how closely the real and virtual actually match. Without feedback, it is difficult to build a system that achieves perfect matches. However, video-based approaches can use image processing or computer vision techniques to aid registration. Since video-based AR systems have a digitized image of the real environment, it may be possible to detect features in the environment and use those to enforce registration. They call this a "closed-loop" approach, since the digitized image provides a mechanism for bringing feedback into the system. In some AR applications it is acceptable to place fiducials in the environment. These fiducials may be LEDs (Light Emitting Diodes) or special markers. The locations or patterns of the fiducials are assumed to be known. Image processing detects the locations of the fiducials, then those are used to make corrections that enforce proper registration. These routines assume that one or more fiducials are visible at all times; without them, the registration can fall apart. But when the fiducials are visible, the results can be accurate to one pixel, which is as close as one can get with video techniques. Instead of fiducials, template matching could be used to achieve registration. Template images of the real objects are taken from a variety of viewpoints. These are used to search the digitized image for the real object. Once that is found, a virtual wireframe can be superimposed on the real object. Recent approaches in video-based matching avoid the need for any calibration. Another approach could be extracting contours from the video of the real world, then use optimization techniques to match the contours of the rendered 3-D virtual object with the contour extracted from the video. Note that calibration-free approaches may not recover all the information required to perform all potential AR tasks. For example, these two approaches do not recover true depth information, which is useful when compositing the real and the virtual. Techniques that use fiducials as the sole tracking source determine the relative projective relationship between the objects in the environment and the video camera. While this is enough to ensure registration, it does not provide all the information one might need in some AR applications, such as the absolute (rather than relative) locations of the objects and the camera. Absolute locations are needed to include virtual and real objects that are not tracked by the video camera, such as a 3-D pointer or other virtual objects not directly tied to real objects in the scene. Additional sensors besides video cameras can aid registration. Laser rangefinders can acquire an initial depth map of the real object in the environment. Given a matching virtual model, the system can match the depth maps from the real and virtual until they are properly aligned, and that provides the information needed for registration.

2.3 Previously Implemented Interfaces for Space Application

Current methods of information management in the space suit are largely unchanged from those used during the Apollo lunar missions of 1969-1972. A small booklet of emergency procedures (Figure 2.6) is mounted on the left arm of the space suit.

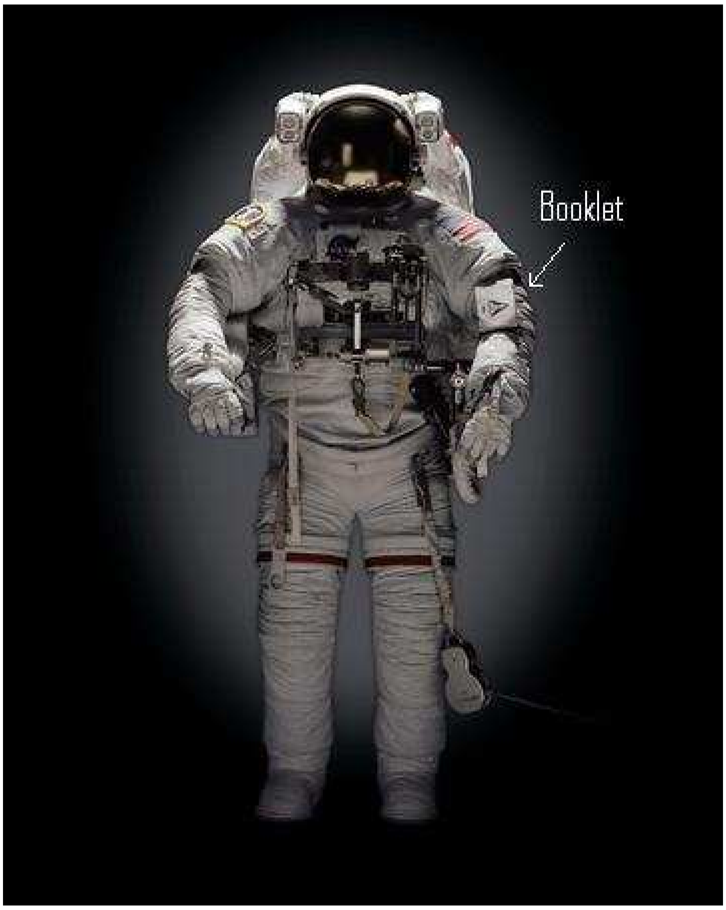


Figure 2.6: Small booklet of emergency procedures mounted on the EMU left arm. (courtesy of James Blair)

Control of radio communications and monitoring of space suit life support functions is accomplished using the display and control module (DCM) on the front of the suit, which includes a small alphanumeric display. The communications carrier assembly (CCA), a headset with redundant noisecanceling microphones, enables hands-free voice communications. Intra-vehicular activity (IVA) astronauts or ground personnel help choreograph EVA activities by communicating each step in a task sequence over the radio to the EVA astronauts. Benefits of suit-accessible hands-free information access and a visual information display were recognized by NASA in the 1980s, when a voice activated computer system with a helmet mounted display (HMD) was proposed for extravehicular activity [8] and a prototype system was developed [9]. This system included a suitexternal HMD that achieved 320 by 220 resolution but suffered from high power consumption (45+ watts versus the EMU total of 58 watts) and field-of-view obstructions. Three additional HMD designs were subsequently developed but none of the four designs was considered for implementation because of great increases in packaging required to incorporate each design into the low profile helmet, protective visor, and solar visor subassemblies of the EMU [10]. A prototype electronic cuff checklist (Figure 2.7) was later developed and flown during four Shuttle flights, but problems of glare, lack of contrast, small font size, cold intolerance, and work envelope interference were noted [11] [7].

Subsequently some attention was devoted to IVA AR wireless visors [6] for robot control and teleoperations, but technological implementation challenges lead to the abandonment of the concept.

2.4 SSL In-House Software Suites

The Space Systems laboratory at the University of Marlyand has developed several software suites in the past that allow communications, monitoring and control of the various systems developed. The MX-2 is equipped with an on board computer (Mac-Mini) that allows the execution of these applications. The aim of this section is to introduce these programs since the AR software implemented in this research effort will be using them.

2.4.1 DMU: Data Management Unit and the HUB

The DMU allows us to acquire and convert analogue readings from all the sensors in the MX-2 spacesuit analogue in digital data. These readings are fundamental for all operations since they allow the support team to monitor the subject and the suit status at all times. The sensors in the suit primarily monitor pressures of: suit, ambient, electronics box, emergency air supplies, and the pneumatic seal.

Figure 2.7: NASAs electronic cuff checklist prototype using a liquid crystal display and touch screen control interface. (courtesy of NASA)

The MX-2 is also equipped with a CO2 sensor and a heart rate monitor that help assess the subject's physical workload and health. The DMU in particular acquires the voltage outputs from the sensors and then through the RCL passes the raw data to the HUB that distributes it through the network. Both the DMU and the HUB are run on the embedded computer in the MX-2.

2.4.2 CS: Control Station

The CS is the support team application for monitoring the MX-2 during operations. This application can be run on any machine that is on the SSL network. It acquires the raw data from the HUB and converts the voltage readings in adequate physical units of the quantity the specific sensor is reading. The support team uses the CS to monitor that sensor readings are within nominal ranges. Experimental applications of the CS allow alternative in-suit CS data access to the test subject through voice commands.

2.4.3 RCL: Ranger Communications Layer

The Ranger Communication Layer (RCL) function is to interface programs enabling them to pass data to one another. The MX-2 System utilizes the RCL to send and receive commands and data thought the SSL network. In particular the MX-2 broadcasts the suit status parameters such as system pressures, subject's heart rate, CO2 readings, etc. MX-2 also transmits and receives voice-recognition commands through this system. The AR software suite uses the RCL to communicate to the suit's DMU and access all suit parameters and voice commands. The use of the RCL will be fundamental in the future when the AR software suite will be able to, for example access data stored on a server and respond to voice commands. The possibilities are endless. RCL technology allows extreme flexibility in future interfaces and AR features implementation without requiring fundamental modifications or redesign of currently implemented software applications.

2.5 ARToolkit

ARToolKit is a C and C++ language software library that lets programmers easily develop Augmented Reality applications. Augmented Reality (AR) is the overlay of virtual computer graphics images on the real world, and has many potential applications in industrial and academic research. One of the most difficult parts of developing an Augmented Reality application is precisely calculating the user's viewpoint in real time so that the virtual images are exactly aligned with real world objects. ARToolKit uses computer vision techniques to calculate the real camera position and orientation relative to marked cards, allowing the programmer to overlay virtual objects onto these cards. The fast, precise tracking provided by ARToolKit should enable the rapid development of many new and interesting AR applications. ARToolKit includes features such as:

- Simple framework for creating real-time augmented reality applications
- Multiplatform library (Windows, Linux, Mac OS X, SGI)
- Overlays 3D virtual objects on real markers (based on computer vision algorithm)
- Multi platform video library with:
  - Multiple input sources (USB, Firewire, capture card) supported
  - Multiple format (RGB/YUV420P, YUV) supported
  - Multiple camera tracking supported
  - GUI initalizing interface
- Fast and cheap 6D marker tracking (real-time planar detection)
- Easy calibration routine
- Simple graphic library (based on GLUT)
- Fast rendering based on OpenGL
- 3D VRML support
- Simple and modular API (in C)

- Other language supported (JAVA, Matlab)
- Complete set of samples and utilities
- OpenSource with GPL license for non-commercial usage

ARToolkit source: (http://www.hitl.washington.edu/artoolkit/) This collection of libraries is open source and is avaliable for free on the web.

2.5.1 Computer Vision Algorithm

ARToolKit is based on a basic corner detection approach with a fast pose estimation algorithm. The ARToolKit tracking works as follows:

1. The camera captures video of the real world and sends it to the computer.
2. Software on the computer searches through each video frame for any square shapes.
3. If a square is found, the software calculates the position of the camera relative to the black square.
4. Once the position of the camera is known a computer graphics model is drawn from that same position.
5. This model is drawn on top of the video of the real world and so appears stuck on the square marker.
6. The final output is shown back on the display, so when the user looks through the display they see graphics overlaid on the real world.

Figures 2.8 , 2.9 and 2.10 summarize these steps. ARToolKit is able to perform this camera tracking in real time, ensuring that the virtual objects always appear overlaid on the tracking markers.

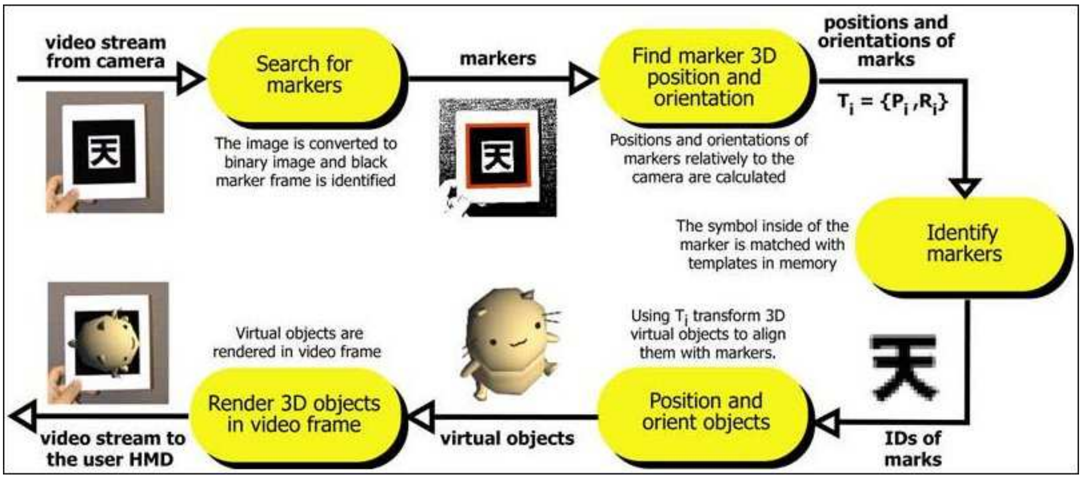


Figure 2.8: ARToolkit acquisition process

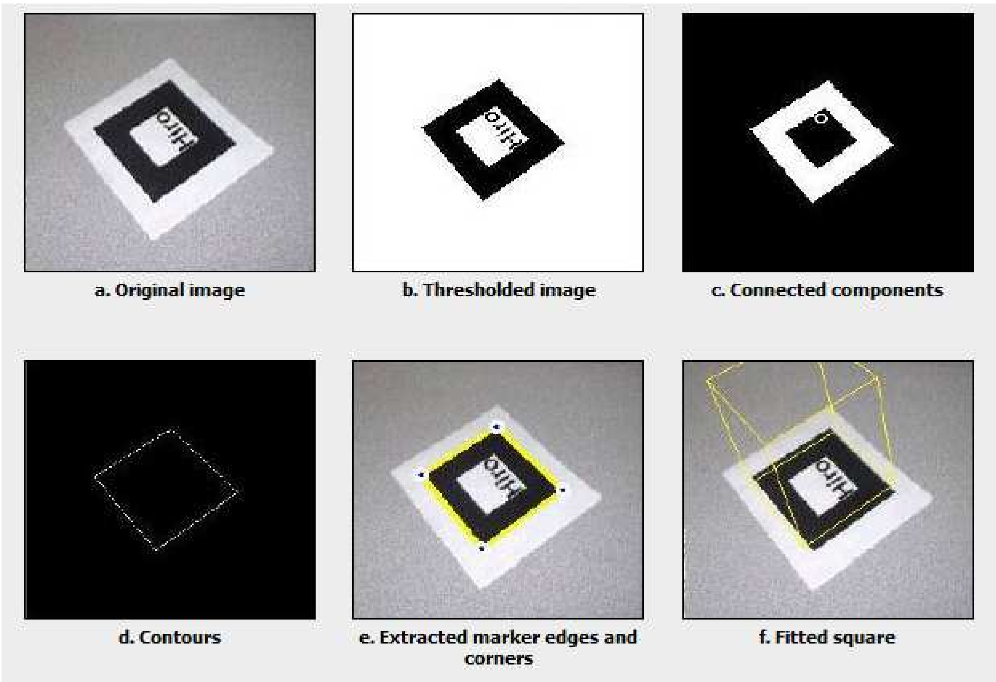


Figure 2.9: ARToolkit marker identification

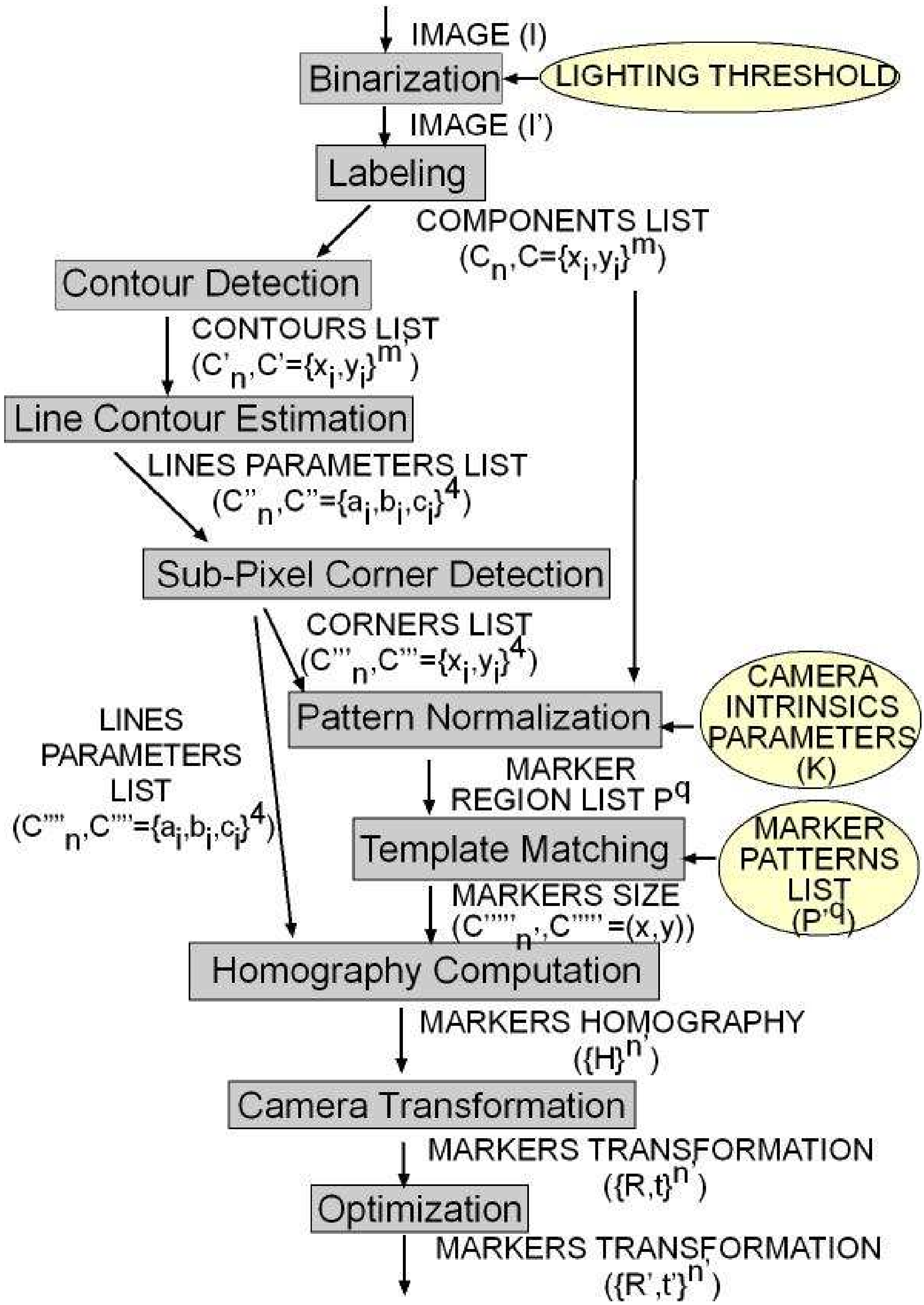


Figure 2.10: ARToolkit algorithm

2.5.2 Limitations of AR Systems

There are some limitations to purely computer vision based AR systems. Naturally the virtual objects will only appear when the tracking marks are in view. This may limit the size or movement of the virtual objects. It also means that if users cover up part of the pattern with their hands or other objects the virtual object will disappear. There are also range issues. The larger the physical pattern the further away the pattern can be detected and so the great volume the user can be tracked in. Table 2.1 shows some typical maximum ranges for square markers of different sizes. These results were gathered by making marker patterns of a range of different sizes (length on a side), placing them perpendicular to the camera and moving the camera back until the virtual objects on the squares disappeared.

Pattern Size (inches)	Usable Maximum Distance (inches)
2.75	16
3.50	25
4.25	34
7.37	50

Table 2.1: Tracking range for different sized patterns

This range is also affected somewhat by pattern complexity. The simpler the pattern the better. Patterns with large black and white regions (i.e. low frequency patterns) are the most effective. Replacing the 4.25 inch square pattern used above, with a pattern of the same size but much more complexity, reduced the tracking range from 34 to 15 inches. Tracking is also affected by the marker orientation relative to the camera. As the markers become more tilted and horizontal, less and less of the center patterns are visible and so the recognition becomes more unreliable. Finally, the tracking results are also affected by lighting conditions. Overhead lights may create reflections and glare spots on a paper marker and so make it more difficult to find the marker square. To reduce the glare, patterns can be made from more non-reflective material. For example, by gluing black velvet fabric to a white base.

2.5.3 Benchmarks

In order to evaluate accuracy of the marker detection, detected position and pose were recorded while the square marker with 80[mm] of side length was moved in depth direction with some slants. Figure 2.11 shows errors of position. Figure 2.12 shows detected slant. This result shows that accuracy decreases the farther the cards are from the camera [3].

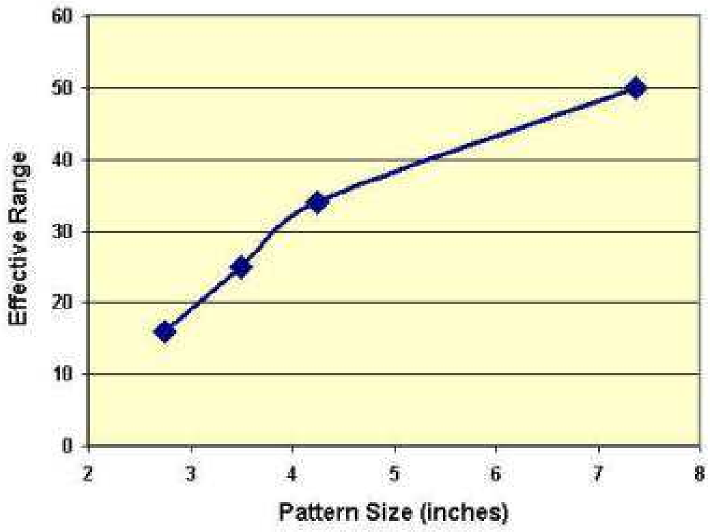


Figure 2.11: Effective Range vs Pattern Size

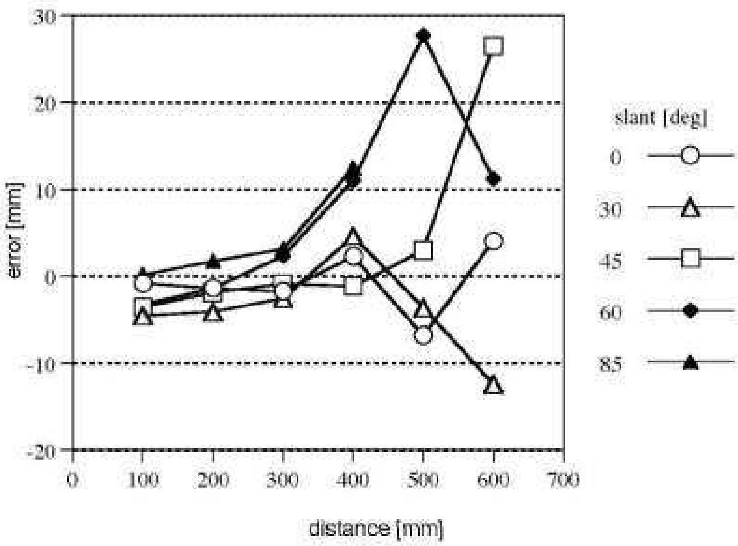


Figure 2.12: Tracking Offset Error vs Range

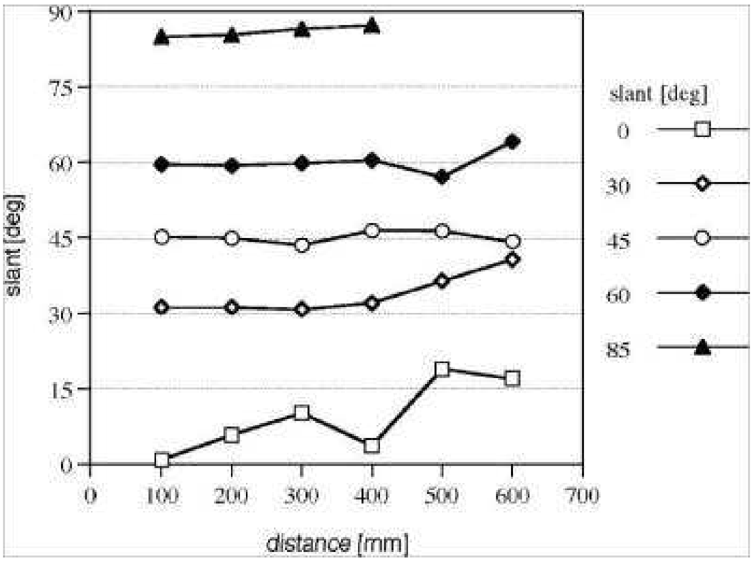


Figure 2.13: Tracking Offset Error vs Angle

2.6 Speed versus Accuracy Decisions in Task Performance

Previous studies adress that subject's regulatory focus influences speed versus accuracy decisions in different tasks. According to the regulatory focus theory, promotion focus concerns with accomplishments and aspirations produce strategic eagerness whereas prevention focus concerns with safety and responsibilities produce strategic vigilance. It has been shown that faster performance and less accuracy in simple tasks were achieved for participants with a chronic or situationally induced promotion focus when compared to participants with prevention focus. It was also shown that as participants move closer to the goal of completing the task, speed increases and accuracy decreases for participants with a promotion focus, whereas speed decreases and accuracy increases for participants with a prevention focus. A promotion focus leads to faster proofreading compared to a pre- vention focus, whereas a prevention focus leads to higher accuracy in finding more difficult errors than a promotion focus. Through speed and searching for easy errors, promotion focus subjects maximize proofreading performance.It is shown that speed/accuracy (or quantity/quality) decisions are influenced by the strategic inclinations of participants varying in regulatory focus rather than by a built-in trade-off[18]

Chapter 3
HARDWARE AND SOFTWARE DEVELOPMENT

In this chapter the hardware development phase of the HUD, the HMD and the digital pin-hole board will be described in detail. The main philosophy behind the development, was the use of available off-the-shelf components which lead to modifications and combinations of the previous. The first system is an LCD based Head Up Display built for integration in the MX-2 spacesuit analogue of the University of Maryland's Space Systems Laboratory. This HUD was very useful for the development of the HMD beacuse initial experimentation identified important features and details that needed to be considered in future designs. The second system is based on a see through Head Mounted Display. This upgraded system has a different typology from the HUD but there are many common features. This chapter is divided in three main sections respectively associated to each system. Each section is then divided in subsections describing: requirements, system development, design considerations, and software developed. Appendices will be referenced for hardware specifications data sheets and software source code.

3.1 HUD:

The first attempt made was to build a Head Up Display that would use an LCD screen because of its partial transparency, small packaging and low power requirements.An off the shelf LCD screen was selected for implementation. Due to availability a 4" LUMIX LCM 480234GF-40CF screen was used. Specifications of the LCD can be found in Appendix A.

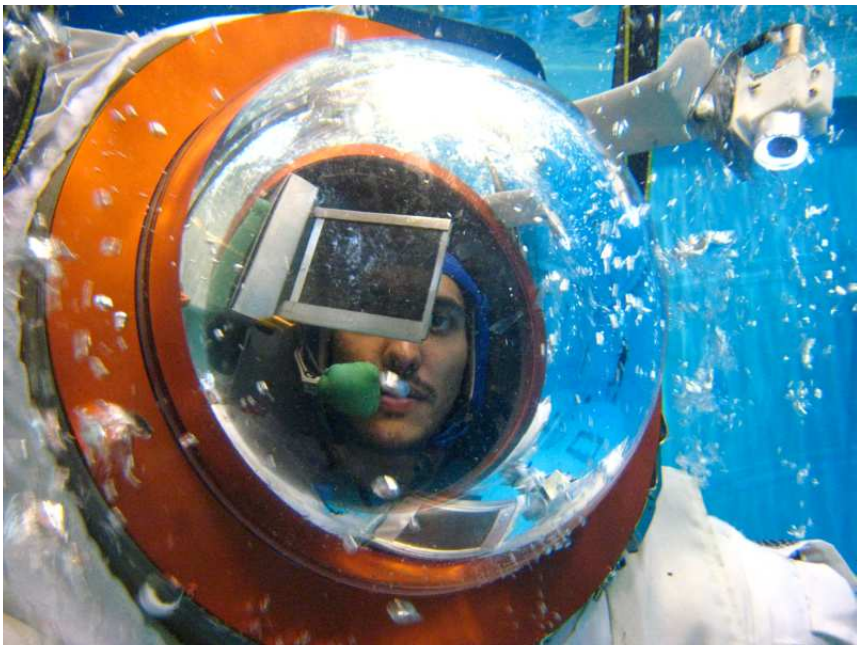


Figure 3.1: HUD in the MX-2 Spaceuit Analogue (UMD SSL, 2008)

This simple and robust interface has been a great source of information for developing increasingly complex applications of In-Suit Digital displays, The HUD has undergone qualitative testing in the latest MX-2 operations, and rapidly demonstrated that it could be a very useful, non critical interface. Although it is by far not an optimal solution for displaying digital information, its testing has defined highly desirable features that should be introduced in the future spacesuit environments.

3.1.1 Design Requirements

Design requirement for the system are as follows:

- The system shall fit in the current MX-2 helmet assembly without interfering with installed equipment and with the subject's head workspace
- The system shall be non-critical for operations purposes.
- The system shall have small power requirements.
- The system must be located outside the subject's main workspace field of view.
- The system must not be a source of electrical or mechanical hazard.
- Wires, connectors and casing shall allow easy removal and servicing.
- The system shall be integrated with the current MX-2 systems.

3.1.2 System Development

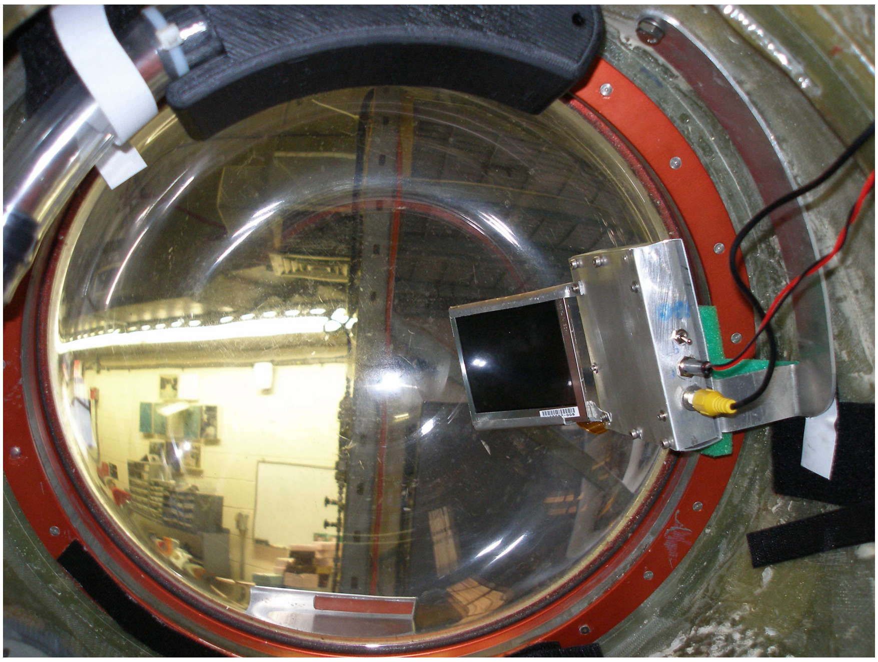


Figure 3.2: HUD Assembly in the MX-2 Spaceuit Analogue (UMD SSL, 2008)

The initial purpose of the HUD was to enable the test subject in the suit to have hands-free access specific task to information such as checklists and diagrams. The system was developed by stripping down the original LCD screen. The back-lamp was removed in order to clear the back of the screen, allowing the user to see through it. The electronics that were originally mounted behind the back-lamp were relocated on the side of the LCD and finally, the original mount was removed and a new mount was designed and built. The mount is divided in three main sections:

- The electronics casing:

The electronics casing is an aluminum box divided in two halves. The bottom half is rigidly connected to the L mount bracket through two screws and it is attached to the top half by four screws. The top half is rigidly connected to the LCD electronics by three pass-through screws. The electronics and the casing are not in direct contact in order to avoid short circuit paths due to the metal casing, therefore a separation layer was introduced between the two elements. The separation layer is composed of two elements: An insulation layer and a vibration damping medium. Insulation is achieved by coating the bottom of the electronics board with electric insulation tape, while vibration damping is achieved through a 1/4 in thick soft foam layer. The foam is then positioned between the board and the metal casing and it is held in place by mechanical pressure induced by the three mounting pass-through screws. The top half also incorporates the power and video-in connectors, an on/off switch and the attachment points for the LCD frame mounts. The bottom half includes a pass-through hole for the video feed ribbon connector from the electronics board to the LCD.

- The L mount bracket

The L mount bracket allows the HUD to be mounted inside the MX-2 helmet. It is attached to the suit's HUT (Hard Upper Torso) through the drinking bag mount screw. Due to mount points availability the HUD had to occupy the drink bag mount, therefore relocation of the previous was necessary. An additional drink bag mount screw was introduced on the L mount bracket, allowing the system to coexist with the drink bag. The mount is a curved L shaped aluminum beam, that runs on the right side of the HUT helmet section with a velcro lip on the bottom. The curvature of the beam is slightly bigger than the HUT's allowing the mount to be pressure fit in the HUT when the mount screw is tightened. A velcro lip was also added to avoid accidental rotation of the mount and it anchors on the drink bag bottom velcro restraint. The L mount bracket is then connected to the HUT and required to be bent in order to locate it inside the helmet assembly. Although fairly stiff, it could still allow the HUD to vibrate and impact on the helmet bubble. For this reason, foam padding has been added on the back side of the mount in order to dump such vibrations and provide adequate separation between the two.

- The LCD frame support

The LCD frame is composed of two U shaped aluminum beams in which the LCD frame is pressure fit. These two beams are each connected through a single screw to the top half of the electronics casing. The two beams are also bent inwards in order to reduce the viewing angle of the screen, and to avoid impact of the LCD edges on the helmet bubble.

In order to satisfy design requirements, all hardware edges were rounded to avoid accidental injury of the test subject. An element of concern was also a particular HUD failure scenario that induced the LCD to shatter. The LCD walls are made of glass that due to the fragile nature of the material could induce small pieces of collide with the test subject's face potentially causing severe injury. On this matter research was conducted to assess the risk involved in this scenario. After analyzing the LCD screen it was noticed that the screen is coated with a plastic adhesive layer that bonds the two sides of the screen avoiding fragment diffusion in the event of a screen fracture. Power requirements were also satisfied since the nominal power consumption of the original screen was estimated to be 5 W, which included the electronics, LCD and backlamp. Since the backlamp was removed, we can safely assess, although it was not measured, that power consumption has been reduced.

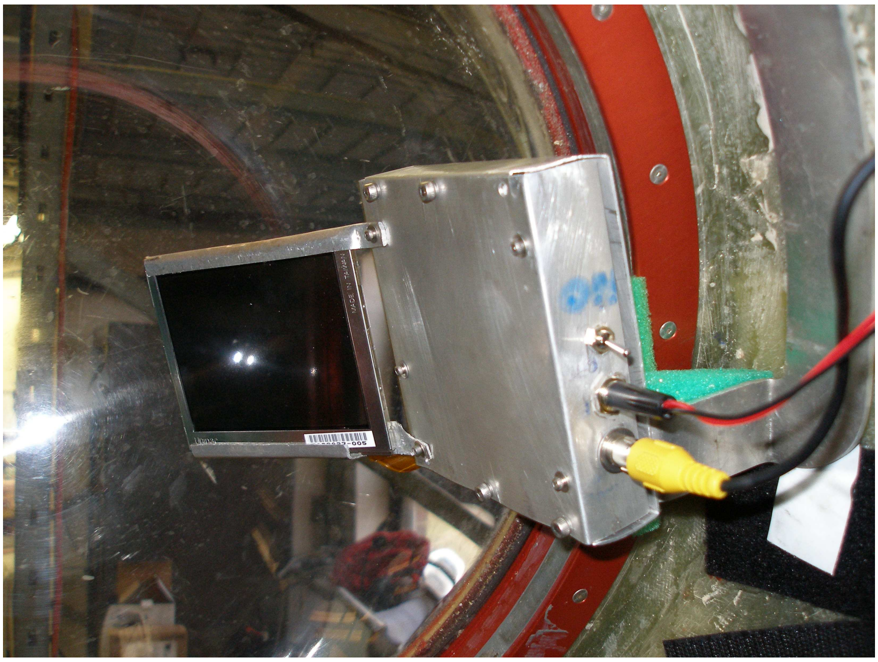


Figure 3.3: HUD Assembly in the MX-2 Spaceuit Analogue Closeup (UMD SSL, 2008)

As part of the integration process, in order to connect the HUD to the electronics in the MX-2 electronics box, a pass through connector was necessary. The first pass through connector was designed to pass: Power (17.5V DC), video feed (NTSC composite), and also incorporated several signal channels cables for future interfaces. The pass through connector is a hollow double side threaded plug in which the cables were passed and kept in place by filling the empty space with epoxy. The filling was necessary since the pass through would connect two section of the suit which have to be kept isolated due to different pressurization. The pass through mount is located just below the communications. connector on the suit entry door assembly, therefore the wires coming from the plug to the HUD had to be routed inside the HUT in order to avoid interference with the test subject. The routing of the cables was achieved through velcro pads that hold together and in place the wires while they pass through the suit. Inside the electronics box the wires are connected as follows: Power is attached to the main power line mammoth connector that feeds 17.5V DC, while the video signal is connected to the SVGA-to-Composite adapter connected to the suit's Mac-mini.

The system integrates in the MX-2 operations procedures easily since it just requires to be turned on through the switch mounted on the back of the electronics casing before the test subject gets in the suit, and it requires to be turned off once operations are complete.

3.1.3 Design Considerations

Qualitative testing of the system [12] revealed much useful information such as:

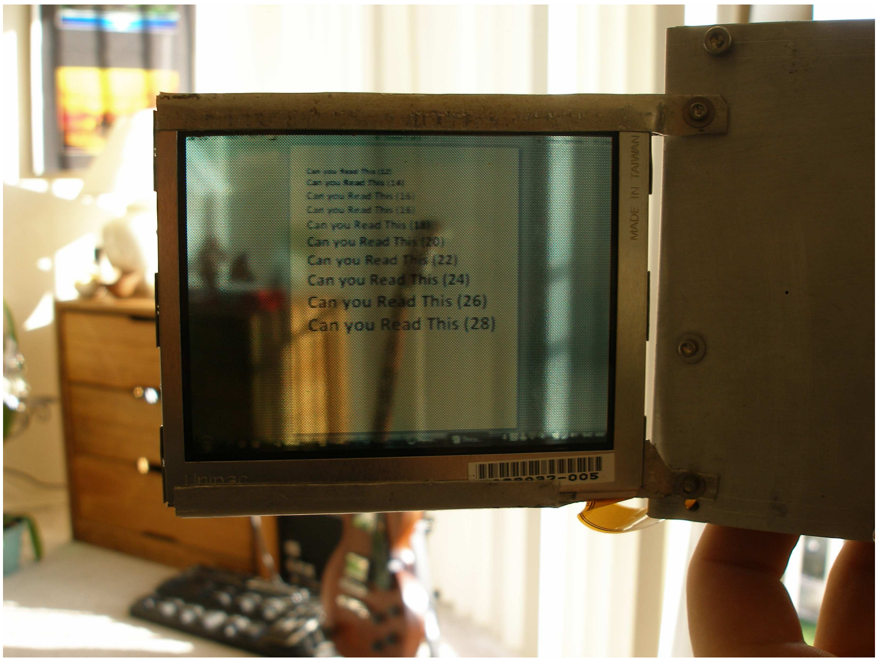


Figure 3.4: HUD displaying text and font sizes.

- Color, Contrast and Pixelation:

During testing in the tank, it was noticed that there are several colors which are practically unusable such as cyan, yellow and orange. In the original LCD setup, the backlamp provided an homogeneous source of white light that increased contrast on the LCD. Its removal induced a large reduction of contrast capabilities therefore dim colors became indistinguishable. This result limits the capabilities of the HUD especially when colors are required for task information such as checklists that include wire color coding or video display. Also the low resolution of the screen and the dimensions of it delivered a displayed image in which each pixel could be distinguished. Images of diagrams that were displayed were poor in quality and looked blurry.

- Proximity to the subject's eyes

Due to the relatively confined available space in the MX-2 helmet, the HUD had to be positioned relatively close to the subject's eyes (approximate distance: 10-12 cm, while a comfortable distance would be around 18-20 cm), making it hard to focus on. Although the HUD was always readable by the subject, it was noticed during testing that whenever information had to be accessed through the HUD the subject would pull their head back to increase the distance. This comfort factor cannot be disregarded for two main reasons: The first is that the MX-2 Helmet is much larger than current spacesuit helmets therefore moving the head back is an option for MX-2 operations but in general this cannot be guaranteed. Second, reading the HUD would require the subject to re-focus on a very different distance, but in this case also to change radically their focus of attention. This would cause a temporary reduction in situational awareness, and would not satisfy the main assumption behind this study.

- Transparency

Although the LCD screen is partially transparent (approximatley 10 % transparent), during testing we noticed that the subjects would hardly try to look at objects through the HUD. The main reasons lies in the fact that the LCD alters colors and the sharpness of the objects behind it, making it very uncomfortable to look through. Details are hidden and shapes are blurred. This effect was noticed early in the development phase and it was the main rationale behind the introduction of the "out-of the main workspace field-of-view" requirement.

3.1.4 Software Development

Software wasn't specifically designed for the HUD since it integrates in the standard computer video output capabilities. However, applications that benefitted from the HUD's capabilities, were many. Just to mention a few: Voice recognition, allowed to display multimedia content on the HUD on command, such as Checklists, Data sheets, Diagrams, video streams, etc. The test subject was also able to access the suit's CS (Control Station), allowing them to know the real-time status of the suit.

3.2 HMD:

Figure 3.5: Final HMD assembly on Support (UMD SSL, 2008)

The HMD system was developed after the HUD in order to compensate on some of the drawbacks of the first system such as: Poor transparency, uncomfortable eye relief, limited color and contrast display capability. Also, research done in the HUD development pointed to several very desirable features that we wanted to implement in the new system including: Augmented Reality and in field-of-view information display techniques. These new features required a different approach on the system design. The monitor based interface approach was abandoned and an optical see-through design was used instead. This approach required the use of optical combiners and video input devices in a Head Mounted Display (HMD) assembly.

3.2.1 Design Requirements

Design requirements for the system are as follows:

- The system shall fit in the current MX-2 helmet assembly without interfering with installed equipment and with the subject's head workspace
- The system shall be non-critical for operations purposes.
- The system shall have small power requirements.
- The system shall be light weight and shall not induce discomfort to the subject.
- The system must not be a source of electrical or mechanical hazard.
- Wires, connectors and casing shall allow easy removal and servicing.
- The system shall be integrated with the current MX-2 systems.
- The system shall be as transparent as possible, and shall not compromise operations that do not require the use of the specific system.
- The system shall be capable of recognizing and tracking targets for AR purposes

3.2.2 System Development

Due to availability, a SONY Glasstron PLM-S600 HMD system was selected. Additional information on the original system can be found in Appendix A. The HMD was initially tested and it was noted that modification to the original system was necessary. The first thing that was noted, was that the HMD was both AR and VR capable. This was achieved through a secondary black and white LCD mounted after the optical combiners that could be darkened or lightened by the user through the HMD integrated control module. This flexible LCD was not necessary for the application we were interested in and it also caused a reduction in the transparency characteristics of the system. An additional layer meant to prevent dust and mechanical damage to the HMD optics was also present and participated in reducing transparency. While testing the HUD we understood the importance of transparency therefore it was chosen to remove both layers. This resulted in a

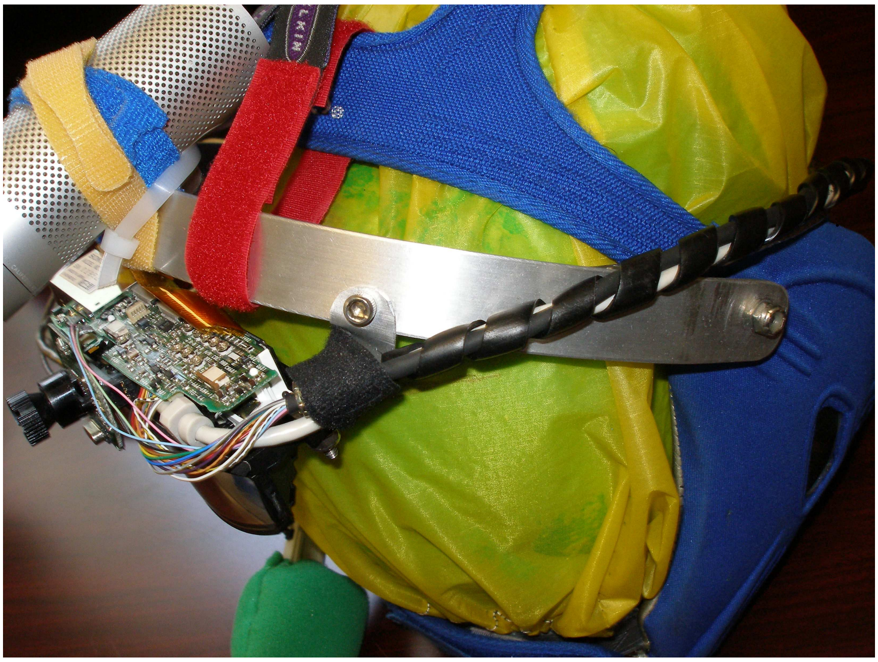


Figure 3.6: Final HMD assembly electronics and mount colseup (UMD SSL, 2008)

significant, although still not optimal increase in the transparency characteristics of the system.

Further preliminary testing of the system also revealed that the original casing of the HMD was designed to reduce glare on the optical combiners but as a drawback, it inhibited the peripheral vision of the subject. Due to our specific application, we had to keep in mind comfort and claustrophobic effects of the HMD while worn in a constrained environment such as the MX-2 spacesuit analogue. In order to regain peripheral vision, the HMD assembly was removed. The system was also separated from the integrated headphones and head restraint, therefore requiring the development of a new mount. The new mount was designed to be fitted on the MX-2 comm. system assembly (snoopy-cap) and it is composed of an aluminum curved beam and two support brackets. The support is then mounted on the snoopy-cap through two screws positioned on the upper headphones assembly. The support was intentionally left free to rotate on the snoopy-cap for personal adjustment reasons, and it is kept in place by two adjustable velcro strips. Further modifications to the HMD were necessary in order to implement a video input device. On this matter, several solutions were developed. The first attempt , was to mount a webcam on the HMD original casing, but due to the subsequent removal of the casing, it was necessary to relocate it. Later, a small spider aluminum mount was built and rigidly mounted to the webcam electronics. This mount was then attached to the HMD assembly and it used the three legs to adjust the camera alignment. Unfortunately the webcam used (GENERAL ELECTRIC EasyCam, further information can be found in Appendix A) delivered very poor performance when connected to the Mac-mini due to lack of specific drivers for this system. In the preliminary testing of the camera, we were constrained to use a generic driver application (MACAM) for Mac OS-X that didn't enable us to use the full resolution of the webcam and refresh rate. Although the system was functional, its performance was not sufficient for AR related target tracking, therefore a new setup was necessary. The initial camera was chosen due to availability and small packaging, but test results and bechmarks with AR applications lead to the conclusion that a better and more compatible camera had to be used. The final camera chosen was an APPLE i-sight firewire camera. Although this second camera is larger, it delivered acceptable performance, and included very useful features such as autofocus and compatibility with the OS-X system. The i-Sight camera was then mounted to the HMD mount through two velcro strips that enabled the user to align the camera with the HMD.

3.2.3 Design Considerations

Qualitative and quantitative testing of the system generated many interesting considerations that will be useful in further development of the system, addressing the specific points that will require more attention.

- HMD Resolution, interpupillary distance and focus distance

The HMD resolution was an item of concern since the early development and testing of the system. Text was generally hard to read in normal font sizes. Increasing font size increased readability of text at the price of reduced amount of text that could be displayed. On the other hand, graphics lacked in detail and were far from being realistic. Although resolution was an item of concern it was not the main constraint of the system. Interpupillary distance on the other hand was noted to be a more severe problem when combined with focusing distance and synthetic overlays of objects at different distances. The HMD system relies on two separate micro LCD screens that generate the synthetic image that is combined with the real world by the optical combiners. The two identical images are then projected in the user's field of view with a certain separation that allows the user to see a single, in-focus image at about 5 feet of distance. The system is not designed to change the separation of the two images, therefore if the user focuses at any depth other than 5 feet, they will see two separate unfocused synthetic images overlaid on an in-focus view of the real object of attention. This problem could be mitigated but not solved by introducing independent control of the screens and stereoscopy techniques. This would allow the user to see a single but not in-focus image. Variable focus of the synthetic image would be a very desirable feature of future HMD systems although very difficult to achieve, since it would require variable focus optics as well as recognition of what is the focus depth of the user.

- HMD field of view, transparency and optical combiners induced distortions

Field of view is one of the major drawbacks of this specific system. The restricted display area, does not allow to display wide objects or even small object in the near field of view. This characteristic as it will be later seen will influence greatly the testing results, and will require the user to orient their head in order to cover wider objects of attention. The tunnel view effect also reduces the overall immersive experience of the user, and feels unnatural so it also may induce claustrofobic effects. These effects if combined with reduced transparency and induced distortions from the optical combiners, require the user a significant adaptation time. During testing it has been also noted that although the user might have adapted to the system, it is highly suggested to limit the time of use to avoid the appearance of side effects.

- HMD Color Display

- Webcam Resolution, view angle, refresh rate and registration

Video input device performance revealed to be crucial for marker identification and tracking. The higher the resolution of the device, the higher the probability to detect a marker. Resolution also plays a role in the maximum distance at which a marker could be detected, but on the other hand, one has to keep in mind that the higher the resolution, the more time it will take the computer to process said images. More important than resolution is refresh rate. The refresh rate of the camera is the number of pictures that the camera is able to acquire in a second. Generally this is limited to 30 fps on general use webcams that although sufficient for video acquisition and playback purposes, it is not for optical see-through AR applications. The main reason behind this is that if a user is looking at a monitor that is displaying a video, the user attention is devoted to the screen and the subject's head and eyes are not moving. Therefore although the eyes refresh rate is much higher than the video/camera refresh rate, the subject doesn't detect a disturbing delay between frames since the video is and feels synthetic. When we introduce optical see-through AR, the user's mind is tricked to believe that the real and the synthetic worlds coexist, therefore ambiguities in the two are greatly enhanced. As it will be seen in the next chapter, during testing, the subjects were very disturbed by the camera refresh rate since their head and eye movements were much faster than the camera, which often resulted in delayed and non-registered synthetic images. Another very important factor in cameras and HMDs is field of view. Current technology in HMDs and cameras do not allow acquisition and display of fields of view comparable to the human eyes, this limitation induces an additional discrepancy between what the user feels as the real and the synthetic world. Lastly, registration of the HMD with the camera also demanded attention. In the setup that was used in this study, the camera above the HMD had to be registered by the user each time. the purpose of this procedure was to make sure that the camera was looking at the same field of view that the subject was looking at through the HMD. This allowed the virtual images and the real world to be aligned and overlaid. Unfortunately ideal registration was often not achieved due to differences in the subjects eye distance from the HMD, distance from the HMD and the board, etc. The reason behind this was that, the HMD and the camera optics induce distortions in the viewed images that are a function of distance from the object of attention and eye-HMD distance. The HMD was initially calibrated for compensating such distortions under specific circumstances, and although these effects were secondary if compared to the ones previously described, they added not negligible discrepancies.

3.2.4 Software Development

The HMD system enabled the introduction of Augmented Reality technology, therefore software had to be developed in order to recognize, and display registered synthetic content. Several application were developed using ARToolkit as a starting point. As described in the previous chapter, ARToolkit is a collection of C++ libraries that allow us to acquire an image from a video input device, analize it, and detect if any markers were present. Once a marker has been detected, we can display 2D or 3D multimedia on the HMD overlayed with the marker. ARToolkit is an extensive collection of functions, but only a few have been used in this specific applications, including:

- Single marker detection and 3d VRML content display;
- Single marker detection and 2d or GL content display;
- Multiple marker detection and single 2d or GL content display:

In order to integrate the HMD with the MX-2 system, we also had to integrate in our software the ability to access RCL data. This enables us to acquire suit sensors information as well as enabling the user to access data from other systems that broadcast data through this medium. This feature will probably enable in the future the ability of the HMD to display robot status information, birds-eye views of robot mounted cameras, and will increase the supervisory capability of the suited astronaut. The code developed can be found in Appendix B where each application is also described in detail.

3.3 Digital Pin-Hole Board:

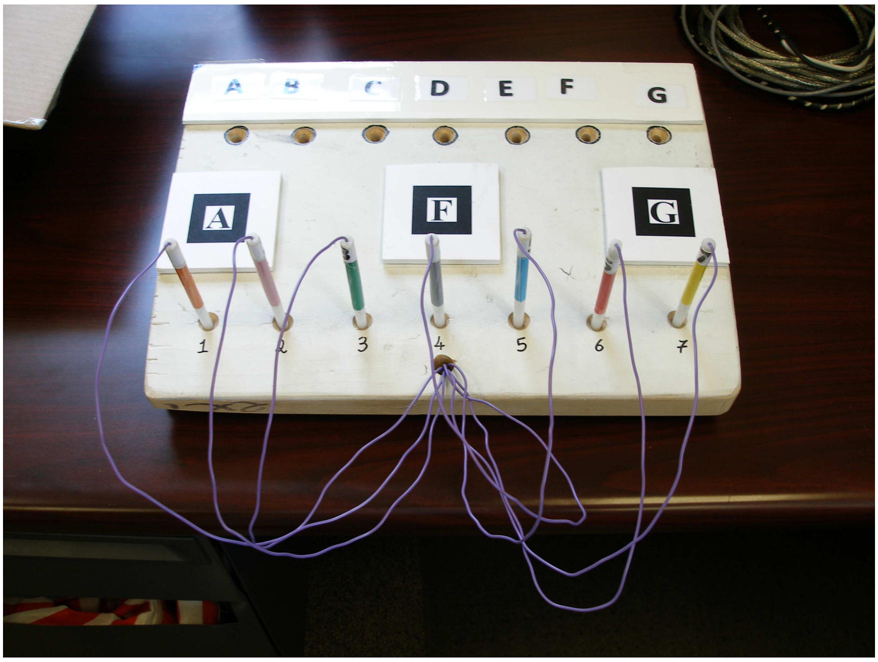


Figure 3.7: Digital Pin Hole Board (UMD SSL, 2008)

The digital pin hole board purpose is to provide a simple task for the test subjects to execute, while allowing the investigator to record execution performance an accuracy automatically. The board was designed to simulate a nominal repetitive EVA task as removing screws from, for example, a plate on the ISS or HST. The system allows variation of the information feed methodology allowing the investigator to study the variations in performance and accuracy in a controlled environment.

3.3.1 Design Requirements

Design requirements are as follows:

- The system shall be capable of recording pin insertions times
- The system shall be capable of determining which pin was inserted in which hole

- the system shall record digital data and shall be compatible with MAC OS-X systems
- The system shall not be a source of mechanical or electrical hazard
- The system shall easily be re-configurable
- The system shall allow task execution without increasing its complexity

3.3.2 System Development

The pin-hole board is based on a 45 x 25 x 12 cm wood tabled, on which 14 holes have been equally distributed in two separate rows. The two horizontal rows are 15 cm apart. Each row has seven holes each 5 cm apart from the others. All the holes have been trimmed to allow easier pin insertion. The board also includes a cable pass through hole in the lower center side. Included in the system are seven acrylic, hollow, cylindrical pins of dimensions 10 cm in length and 5 mm in diameter. A cable is then passed through the pins. On the back of the board, two NI-DAQ USB 6008 cards are placed and connected as follows: Holes 1 through 4 are connected to the analog channels ai0 through ai3 of the first card, while the remaining three holes are connected to the second card. Unfortunately both cards had only 4 working analog channels each, hence the two cards. The pin cables are then connected to the 5V output port of the first A/D card through a voltage divider so that each pin carries a different voltage. In conclusion, by applying different voltages to each pin, and different channels on the A/D card to each hole, we can distinguish which pin has been inserted where.

3.3.3 Design Considerations

- Cables tangling

For the board to record pin insertions, the pins needed to be connected to the electronics on the back of the board. Connection was achieved through a single cable attached to each pin. These cables tangled easily during task execution therefore the task procedure was modified in order to reduce the chances of it affecting the recorded data. Initially the procedure saw the user leaving the pins in the holes once inserted. It was later assessed that the procedure could be changed without

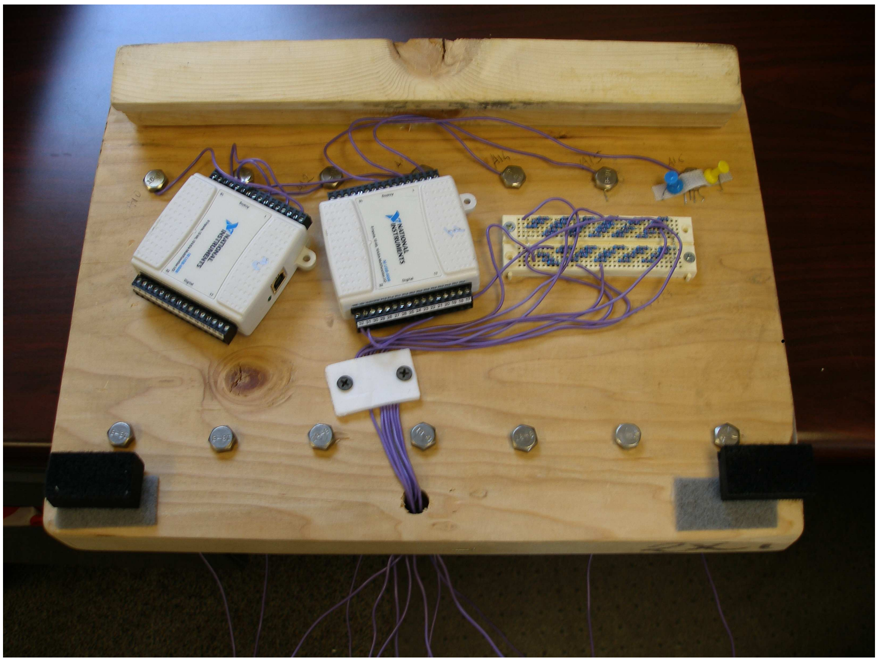


Figure 3.8: Digital Pin Hole Board Electronics (UMD SSL, 2008)

affecting the task by asking the subjects to insert and remove the pins in the holes leaving the pins hanging on the bottom side of the board. Although not optimal this procedure avoided: cables tangling during testing and marker occlusion. Future implementations of the board could see a wire-less pin configuration that would eliminate this problem.

- Holes clogging

Due to the brittle nature of wood, the extensive use of the board caused it to splinter and clog the holes after several pin insertions. This phenomenon did not compromise excessively the data acquisition process, although it required the holes to be cleaned often. Holes were cleaned by blowing in the hole and removing the wood splinters from within.

- Holes identification study

Preliminary testing of the system focused mainly on debugging the pin-hole board software and hardware configuration as well as assessing an effective method of distinguishing pins and holes. Several configurations were tested where the pins were identified by numbers and the holes by first letters, then shapes, and finally colors. Results showed that letters and numbers were the most effective solution since they provided an order reference to the subject, reducing the task complexity.

- Pins reconfiguration after each task

The board, required the subjects to reconfigure the pins after each trial. This task was not timed during the experiment and it was necessary since the board has no capability of determining the initial configuration of the pins.

3.3.4 Software Development

For testing purposes three specific software applications were developed for use with this board. The applications enable us to acquire digital data from the board such as, which pin was inserted where, and when it was inserted. The applications also evaluate if the trials were executed correctly and return a log file that includes all the experiment history. These applications are independent of the type ofthe pin hole identification method used, as long as the reference file is formatted properly. Further details on the specific applications can be found in Appendix B.

Chapter 4

TESTING: Performance and Accuracy Impact on Simple Unrehearsed Tasks

In the previous chapters, it has been seen that the introduction of digital information displays would enable several interesting features for future EVA and IVA. Intrinsic benefits aside, the point of interest of this study is to understand the main features and characteristics of see-through HMDs on which future development should focus in order to increase usability, user response, task performance and accuracy. Unfortunately previous experience on these systems for this particular application is very limited, therefore nothing could be assessed beforehand. In this experiment a simple unrehearsed task was simulated by asking the subject to remove a specific peg and insert it in a different hole on a digital pin-hole board. Similar tasks (although not unrehearsed) can be seen on current EVAs. A good example for this would be the future STS-125 mission, where astronauts will service the Hubble Space Telescope and will have to remove several hundred screws and bolts from the external panels of the telescope before being able to service the internal systems. In the previous example, one could struggle to see similarities since the described task, is rehearsed extensively and assisted through voice communication with Mission Control. But what if a similar task had to be executed a few years from now either on the Lunar or Martian surface or in orbit around Mars to repair an unforeseen system failure? In this case, similarities are easier to see. Voice communication would be impractical and it would be reasonable to assume that the astronauts would not have rehearsed the task extensively beforehand. So given the actual state of EVA systems, the following experiment was designed to compare today's information display systems (paper checklists) with an HMD based system in two different configurations. Relevant aspects in the comparison will be task execution performance and accuracy in the execution. The experiment was divided in three sections in the attempt to isolate and identify these factors:

- Set a baseline. For this purpose, the first test (Paper Cards) was included in order to measure: The time it takes a subject to find and focus on the deck of cards; Find, read and understand a specific instruction; Focus on the board;

Find the specific pin and hole, and finally execute the motory task.

- The second Test, was included in order to measure performance when the subject was not required to find and focus on the deck of cards. This was achieved by including an HMD system capable of displaying the task information directly in the user's field of view, while still allowing them to see through it.

- Finally the third test, was designed in order to reduce or ideally eliminate the subject's cognitive processes as: Finding the instruction, reading and understanding it and identifying the pin and the hole. This was achieved by displaying on the HMD a graphical overlay with lines connecting the pins and the holes. In this case, the subject was not required to recognize or read anything, the only thing that they would have to do is to follow the lines.

This experiment was designed to be a within subjects experiment indicating that each subject would execute all three tests, in the same order. In each Test, subjects would use one of the three sets of cards that were generated and the order in which they would use the cards was randomized for each subject. The three sets of cards were equivalent for statistical purposes. The cards and instruction sets will be further described later in this chapter.

4.1 Experiment Hypothesis

The experiment's purpose is to study performance and accuracy benefits if any, resulting from the use of in-field-of-view information display system when executing a simple unrehearsed task. Secondary goals are the identification of fundamental HMD characteristics for future hardware development, as well as studiing the user reponce to the system.

4.2 Experimental Protocol

4.2.1 Test Subject selection

Subjects will be adult students (over 18 years of age), faculty or staff from the College of Engineering of the University of Maryland. The number of subjects in the study will be between 10 and 15. The subjects will not be selected for race, ethnic origin, religion, or any social or economic qualifications. Ineligibility to the study will be caused exclusively from physical impairment that is expected to reduce speed and accuracy in movements. Subjects will be recruited formally through email. The recruitment email will be sent exclusively to the SSL Faculty, Students and Staff. Study participation will be completely voluntary. Subjects can withdraw from the study at any time for any reason.

4.2.2 Method

4.2.2.1 Test 1: Paper Cards

In this first set of trials, the user will receive a small deck of cards, containing the sequences of pin insertions to execute. The deck is made of regular paper and contains 30 cards stapled together in the upper left corner. Each card contains a random set of insertions that vary in number from card to card; from a minimum of 2 insertions to a maximum of 7. Each instruction is displayed as a number indicating the pin identifier followed by an arrow and a letter indicating the hole identifier. In the experiment, the subject will be asked to execute each sequence of insertions from the first to the last in vertical order. Three separate decks named DeckA, DeckB and DeckC were created in matlab by generating two random permutations of numbers from 1 to 7 and ignoring the the last 0 to 5 numbers in each sequence. The number of ignored digits, was random as well but the cards were generated in such a way that the number of insertions per card was equally distributed in each deck. For this reason, each deck contains a number of cards multiple of 6, containing as many 2 pin insertions cards as 3, 4, 5, 6 and 7 pin insertions per cards. The matlab code for generating the cards can be found in Appendix B. Once the subject was introduced to the cards, they were asked to run a minimum of 5 test trials to familiarize with the system. For this practice trial a different set of cards from the one that would be used for data acquisition was used, and performance data were not acquired. Subjects were invited to execute as many familiarization trials as desired. During the familiarization trials, the subject was taught how to execute each task. The subject execution sequence can be summarized as follows:

- The subject was asked to position the deck facing downward on a flat surface in the vicinity of the board. The deck could be positioned anywhere except directly on the board. Notes on where each subject positioned the deck were taken for later comparison and analysis.

- During this experiment a computer was used for recording the subject's performance. Recording was triggered by the user in each trial by pressing the "Enter" key on the keyboard on their side. The user could reposition the keyboard during the familiarization trials in order to maximize confort and ease of use. A note on keyboard position was also taken for later comparison and analysis
- The subject would then verify the pin's initial configuration, then once ready they would flip the deck and simultaneously press the "Enter" key on the keyboard. This action would initiate the timer on the computer, and give access to the trial information to the user.
- Once the deck is flipped, the subject was asked to execute as rapidly and as accurately as possible each set of instructions displayed, starting from the first to the last, by extracting the first indicated pin from the lower part of the board, and inserting it in the upper part. Once the pin is inserted, the computer would recognize the insertion, and return an audio feedback tone indicating that the insertion was recorded.
- After the audio feedback tone is received by the user, they are asked to extract the pin and let it hang from its wire on the lower part of the board.
- Once the first pin insertion is executed, the subject will repeat the insertion sequence until all indicated pins have been inserted.
- After all the indicated pins have been inserted, the timer is stopped and the subject is asked to reposition each pin in its relative hole and flip a page on the deck, without looking, and reposition it face down on the chosen surface.
- Once again, when ready, the subject would repeat the sequence for at least 5 times during familiarization or until the last card in the deck during the real test.

During the test, data was acquired and recorded on a computer in the form of a log file containing the insertion times of each pin, as well as the pin and the hole identifiers in which it was inserted. The log file also included an accuracy check, and returned for each insertion if it was executed correctly of not. Although the system is able to distinguish between correct or incorrect insertions, the user is not made aware of it. Due to the experimental setup, the subjects were also advised on what to do in case they recognized a wrong insertion. To note is that in each trial, each pin and hole could be used only once. The recording system would record a pin insertion, and then disable further readings from the used hole, therefore if the user misplaced a pin, and then attempted to insert a different pin in the same hole, they would not receive any audio warning. In this case, the user was told to skim through the instructions and find a unused hole and insert the pin there. For later analysis, all the data was filtered beforehand by ignoring any data point corresponding to a wrong insertion as well as datapoints larger in magnitude than the mean insertion time for the entire trial plus or minus two standard deviations.

4.2.2.2 Test 2: HMD Virtual Cards

For this part of the experiment, the user was introduced to the HMD system that would display virtual cards within the user field of view. The experiment setup was identical to the previous except for the following details:

- The virtual cards were displayed on the HMD system color inverted as in white text on a black background. The reason behind this choice lies in the HMD display methodology. The HMD displays information by controlling two micro LCD screens that filter the light coming from a small white backlamp. In order to keep the screen transparent, we would have to send a black screen input to the HMD. In this case, the micro LCD's would filter all the light coming from the backlamp and allow the user to see through the optical combiners. Due to this technical detail, since in this part of the experiment we wanted to allow the subjects to see both the instructions and the board, the information display was triggered in order to maximize contrast between the virtual and the real image, hence the inverted color cards.

- In this second test, due to software implementation difficulties, the user would use two keyboards instead of one. The two keyboards would be controlling two differnet computers, one on which the recording software is running and the other where the cards are displayed and sent to the visor. In order to initiate each task, the two keyboards were positioned one above the other and the user was asked to press simultaneously the “Enter” and the “right arrow” key on the two keyboards. This would flip the card page and initiate recording.

4.2.2.3 Test 3: HMD Augmeted Reality Graphical Overlay

In the last part of the experiment, the subjects would use, once again, the HMD system, and although the methodology behind the experiment execution remained essentially the same as in the previous tests, the information display typology was very different. Differences from the previous setups are as follows:

- The HMD will make use of the integrated camera to acquire visual information on the subject’s field of view, it will then recognize, if visible, the fiducials on the board, and overlay a synthetic image indicating which pin to insert where. For the system to know where the board is, only one fiducial is necessary. In this case three fiducial are used for redundancy and due to the fact that the camera cannot see the entire board at once. The three fiducial allow the system to recognize where the board is even if the user/camera can see only a portion of it. Also to note is that multiple fiducials, if detected simultaneously, increase the accuracy in the estimation of the board position. Instructions were displayed in the form of a line connecting the pin to be moved and the hole where it would have to be inserted. The HMD would also display indicators for both the pins and holes in the form of circles.

- Since the HMD does not allow stereoscopic registration, in order to avoid confusion to the user, subjects were asked which was their dominant eye and they were asked to cover the optical combiner on the opposite eye with a small post-it. This would prevent the user from seeing two different sets of lines (instructions) due to the different focal distance between the HMD’s optical combiners and the board.

- Since the HMD configuration and calibration was user dependant, each subject was asked before commencing the familiarization trials, to calibrate the HMD’s camera, by rotating it and registering the virtual overlay on the physical board. Once this was achieved the familiarization test would begin.

- In this scenario, only one keyboard was used to start the timer, and although the methodology was identical to the first test, in this case, the user would see only one instruction at a time. The HMD would display a single line at a time and once an insertion was recognized the next line would appear. Once all the trial insertions were executed, the image would freeze and the user would reset the pins and press “enter” to continue.

4.3 Experimental Results Analysis

In this section the data analysis will be described in detail. The general approach during this phase, was to initially validate the data statistically and then study eventual trends or interesting features. qualitative data was also used in order to determine the semantics behind the results. The experimental data was acquired mainly in two ways:

- Questionnaires : Each test subject, was asked to fill in two questionnaires, one before commencing the experiment and one at the end. In the first questionnaire, the subject, was asked to fill in his personal information, in order to define the subject population that participated in the study. The second questionnaire, on the other hand, was supposed to give the principal investigator feedback on each setup. this data, combined with the performance and accuracy measurements, allowed a much deeper understanding of the scenarios that took place.

- Performance and Accuracy Data : Specific software was developed in order to read from the digital pin-hole board the time at which each pin insertion occurred. Times per each set were acquired between when the enter button on the keyboard was pressed and when the first pin was inserted. After the first insertion of the set, times were recorded between the previous and the next insertion. The accuracy at which the times were acquired was in the order of milliseconds, this was dictated by the accuracy of the computer's internal clock. Pin insertion accuracy was also recorded by appending, in a log file, the pin and hole identifiers and the insertion time.

  In the same log file, each insertion was identified by the following characteristics:

  - Insertion Time
  - Pin #
  - Hole #
  - Accuracy in the form of "Right" or "Wrong" (this parameter was evaluated by comparing the recoded pin and hole identifiers with the desired ones.)

– Difficulty from 1 - 7 (this parameter was evaluated by associating distance that the pin had to be moved with a number from 1 to 7)

Other qualitative data was also acquired during the tests in the form of notes. These notes were taken by the principal investigator, and they mainly focused on the subject's personal procedures, preference in the positioning of the equipment and also relevant comments that were not included in the questionnaires.

4.3.1 Subjects Population

This experiment ideally would have had to be proposed to a population of astronauts, but due to obvious difficulties in gathering such subjects, students within the university's department of Aerospace Engineering were chosen as a suitable analogue. In total 11 subjects participated to the experiment, mostly male graduate students. Interestingly, the female population in this experiment was comparable in percentage to the astronaut core (20-30 %). Within the eleven subjects, only two had some previous experience with HMDs and it was mainly with VR systems for gaming purposes. Previous experience was also very limited, therefore it was far from being a representative parameter for defining a subgroup of subjects. This last consideration was interesting since it pointed out that although the general population is aware of the existence of these devices, they rarely have the opportunity to use them, indicating that even in the everyday life, HMDs are not too popular. Previous experience, age range and reduced vision parameters can be seen as the main differences between this and the ideal population. User preference also pointed out that subjects could be divided in two significant subgroups: people who preferred the physical cards and people who preferred the virtual cards on the HMD. As it will be shown later on in the analysis section, this subgroup division is not plausible and will be explained later. Lastly an important piece of information was given on sideffects. From the questionnaires, it resulted that 8 subjects experienced either headache, eye fatigue or in general disconfort while using the HMD. These sideffects were described as mild and the subjects did not want to terminate the experiment prematurely. Although there was no real trend between subjects preference and the experience of sideffects, it was noted that these were common in subjects who had difficulties adapting to the system. Further information on HMD adaptation was gathered both through notes and through the last section of the second questionnaire which will be expanded later in the chapter.

The following table summarizes the above description and was acquired in the first and part of the second questionnaire.

Subject Population and Setup Preferences.

Subjects	11
Age	21-27
Gender	8 Male, 3 Female
Uni. Affiliation	10 Grad, 1 Undergrad
Left/fight Handed	9 Right, 2 Left
Reduced Vision	7
Corrected Vision	7
Color blind	0
Reduced Arm Mobility	0
Previous HMD experience	2
Experienced Side Effects?	8
Preferred System	
Setup 1	5
Setup 2	5
Setup 3	1

4.3.2 Performance analysis

A preliminary view to the test results is given by the following graph that shows the mean pin insertion time per subject for the three tests.

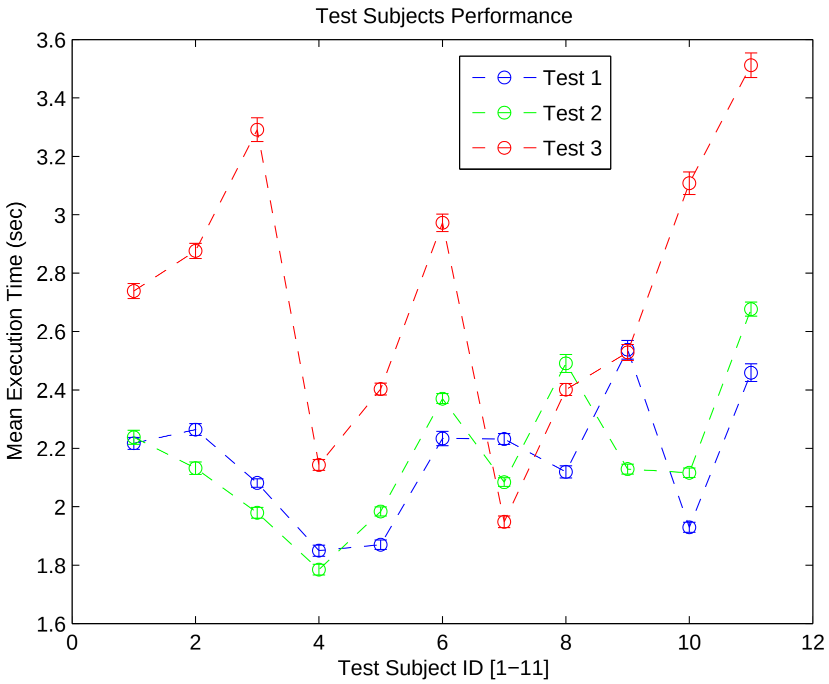


Figure 4.1: Mean Execution Time per Subject per Test

As it can be seen clearly seen in the above graph, the differences in performance are small between the first and second setups, while there is no doubt of statistical difference betwen these two and the third. A second set of results, is the analysis of the time trend of insertion times. The following graph shows, the time trend for each subject, for each test.

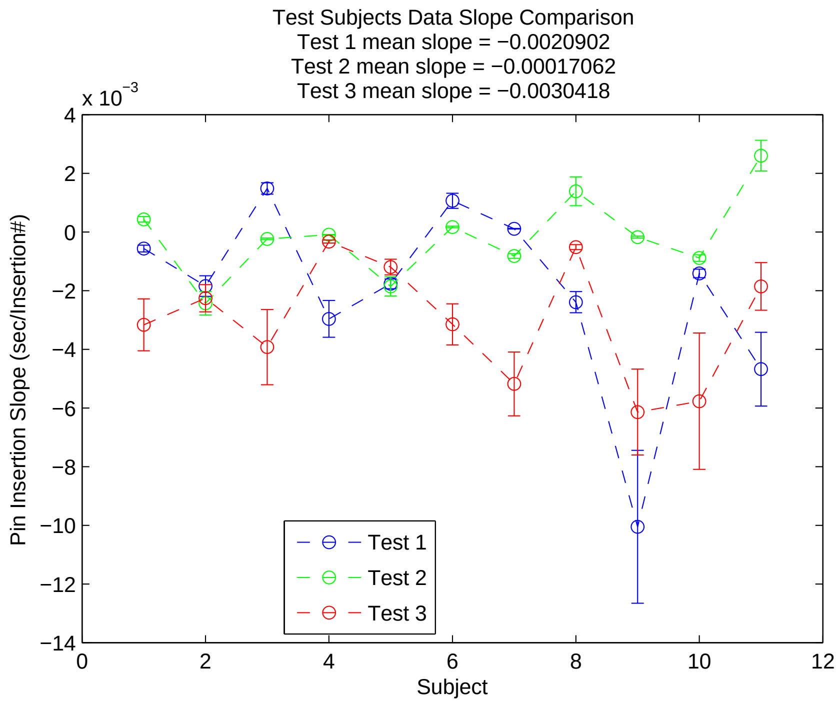


Figure 4.2: Test Subjects Data Slope Comparison

The previous graph was plotted in order to determine if there were any learning or fatigue effects during the tests. This was achieved by calculating the mean slope between data points and insertion order on the entire set of acquired data for each test. As it can be seen, the slopes are all in the order of millisec/insertion #

From this plot, we can safely assess that learning or fatigue effects was negligible. Overall the time increase or decrease due to these effects would not be more than 0.1 sec over the entire duration of each test.

Performance were intially analyzed through SAS (Statistical Analysis Software). This tool was used to determine if the data acquired in the three setups was statistically relevant. SAS was setup to run ANOVA (Analysis of Variance) analysis on a series of scenarios where different sets of data were fed with different statistical models. SAS was used to acquire raw scattered data and return the probability of that data to be dependant on certain parameters. The parameters to be analyzed where defined in the statistical model. Two models have been used, a third order model and a linear model. The reason why it was chosen to run these two models, was to quantify the error that we would make when assuming that any eventual dependand variable was a linear function of time. In order to compare these models, we had to compare the -2 Residual Log Like Parameter, which indicated the accuracy of fit between the model and the data. The assumption behind the comparison is that the third order model, in general is more reliable, therefore if the -2 Residual Log Like Parameter for the two models are comparable, it is reasonable to assume the parameter in question to be a linear function of time. Several scenarios were fed to SAS and results can be summarized as follows:

- Generic Mixed Model SAS Input file:

```
filename resfile '/homes/maxdc/SAS/DataSAS';
    *Read from file;
    data subjects;
    infile resfile DLM=',' FIRSTOBS=2;
    input SubjectID PinPerTask Pin Hole Difficulty TestID Time Correct;
    run;
    *Create model;
    proc mixed data=subjects;
    class PinPerTask Difficulty TestID SubjectID;
    model Time = PinPerTask—Difficulty—TestID ;
    random SubjectID /subject=SubjectID;

    run;
```

- SAS: ANOVA Mixed Model Analysis

The first model fed was:

Time = PinPerTask — Difficulty — TestID

The above syntax in SAS will return the dependency probability of all possible combinations of the PinPerTask, Difficulty and TestID parameters. The analysis was done on 7 different sets of data in order to compare the three test setups. First SAS was input the complete data from all three setups, then the single tests and finally the three combinations. Results are as follows:

SAS: ANOVA Mixed Model Analysis Results

Parameter	All Tests	Test 1	Test 2	Test 3
Cov. SubjectID	0.1270	0.09969	0.1017	0.3264
PinPerTask	0.8482	0.6716	0.5451	0.0864
Difficulty	¡.0001	0.4736	0.0028	0.0007
PinPerTas*Difficulty	0.3197	0.9996	0.1410	0.7523
TestID	¡.0001	–	–	–
PinPerTask*TestID	0.0262	–	–	–
Difficulty*TestID	0.0532	–	–	–
PinPer*Diffic*TestID	0.9599	–	–	–

Parameter	Test 1-2	Test 2-3	Test 1-3
Cov. SubjectID	0.0750	0.1491	0.1128
PinPerTask	0.4750	0.6647	0.7137
Difficulty	0.0159	¡.0001	0.0156
PinPerTas*Difficulty	0.7685	0.1048	0.9921
TestID	0.1206	¡.0001	¡.0001
PinPerTask*TestID	0.8249	0.0383	0.1363
Difficulty*TestID	0.8513	0.1539	0.1272
PinPer*Diffic*TestID	0.9974	0.8840	0.9902

In the above tables, except for the Covariance between SubjectIDs, all the numbers indicate wether the parameter was significant or not. In order to quantify this, SAS examines the model by normalizing the dependant variable (time, from 1 to 0), then analyzes the dependencies of the model when each factor is assumed to be 0. The result shown above is the normalized value of the dependant variable when the relative term is 0. Therefore we can see that the smaller the term, the more the model depends on said variable. The covariance parameter follows a different trend, the bigger the parameter, the greater the differences between subjects. The covariance in this case, is expressed in seconds.

In order to assess that a certain parameter is significant, we would have to define a threshold below which the condition is verified. In this case, the threshold will be a variable parameter, since the aim of this analysis is to identify possible trends.

The first thing we see from the All Tests results is that there are only two significant parameters to consider; the Difficulty and the the Test ID. This second result was expected, since as seen from Graph 4.1 the three tests show significantly different mean values per subject, at least between the first two tests and the third. On the other side the fact that data is dependent on the difficulty parameter, although expected, it was not visible form the previous summary graph. It was also interesting to see that the covariance between subjects is small (0.0127 sec ¡1% of the mean) therefore we can safely asses that the considered population is consistent and statistically significant.

The All Tests analyses did show some interesting results, but it just says that there are differences between the three tests and not wether there are any differences between the tests, or within the tests. For this reason, separate analysis were executed. From these we can say that the Difficulty parameter remains significant at least for test 2 and 3 while it looses significance in test 1. What we can also say is that it remains significant within the combinations of tests. As expected, we can also see that differences between the first / second and the third are significant, but differences between the first and the second can be assessed only within a 12 % uncertainty.

Following the previous results, the Mean execution time per subject were plotted versus the insertion difficulty. As SAS predicted an increasing trend is visible, and appears to be somewhat linear.

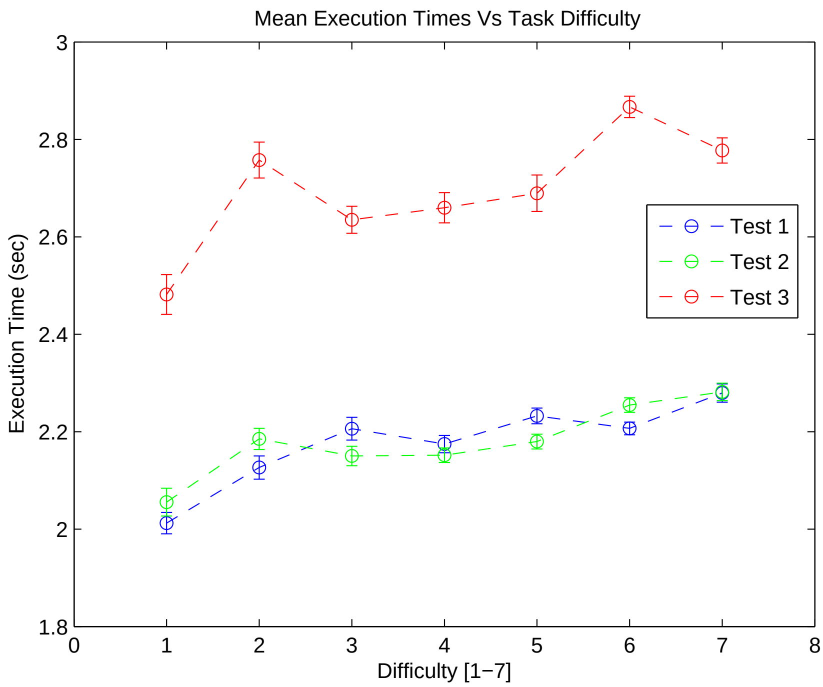


Figure 4.3: Mean Execution Time Vs Pin Insertion Difficulty

- SAS: ANOVA Linear Model Analysis

As previously seen, all the other parameters, especially the higher order terms seem to be negligible. This last point raises a question. Would it be possible to consider the difficulty parameter as a linear function of time? In order to do so, the following model was fed to SAS:

Time = PinPerTask Difficulty TestID

The above syntax in SAS will run the ANOVA analysis with a linear model where only the first order interactions between parameters are considered. Three cases were ran, corresponding to the single tests setups. The results we are interested in comparing in this case are mainly the fit "goodness" parameter or as it is called in SAS; the -2 Res Log Like. If the parameter from the previous and the one from the model fall within a tolerance level usually defined as 5 % we can reasonably assume the trend to be true.

Test	Non-Linear Model	Linear Model	Difference (%)
Test 1	5992.8	6011.7	0.3 %
Test 2	4477.7	4479.2	0.03 %
Test 3	5475.5	5508.2	0.5 %

From the above results we can see that it is reasonable to assume the trend linear, therefore it was evaluated and plotted in the following graph:

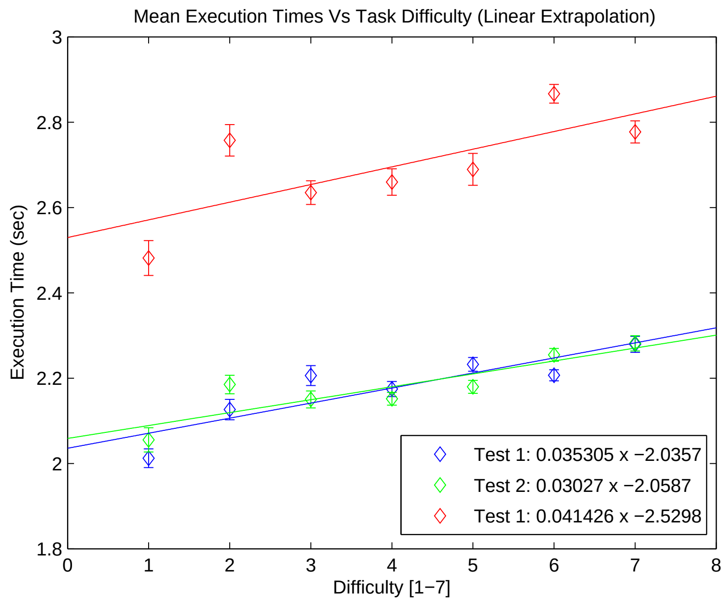


Figure 4.4: Mean Execution Time Vs Pin Insertion Difficulty Slope Extrapolation

Although the apparent trend between the first and second tests, it is not possible to adress any conclusions since the two curves are two close toghether dening any possible distinction. Also the slope of the third setup is higher than both the first and second, therefore it will never intersect the previous resulting in poorer performance constantly. The same results are achieved if insead of plotting the data with the difficulty parameter, we plot it with the actual distance that the pin had to be moved. According to Fitt's law we should see an exponential trend in the insertion time when the distance increases. Unfortunatley the pin movements were confined in a very small region, therefore there it was not possible to detect such trend.

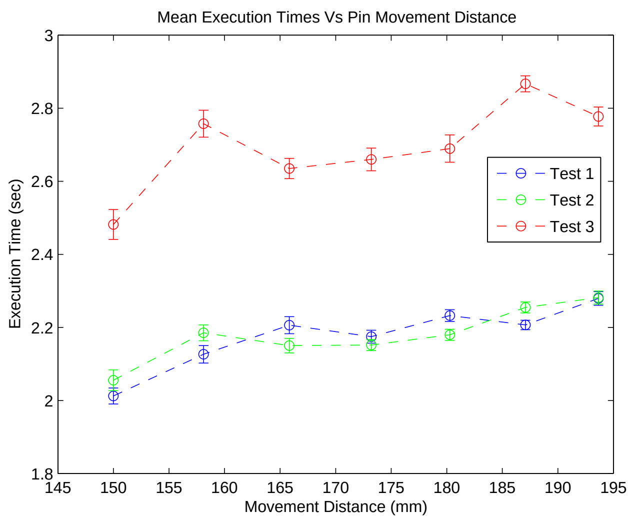


Figure 4.5: Mean Execution Time Vs Pin Insertion Movement Distance

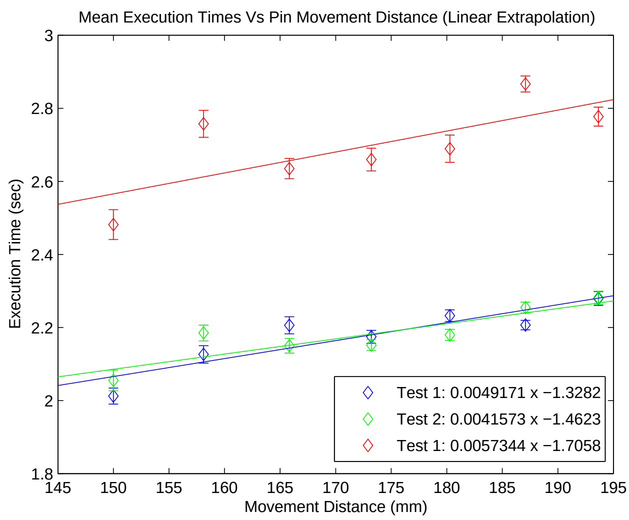


Figure 4.6: Mean Execution Time Vs Pin Insertion Movement Distance Linear Extrapolation

4.3.3 Error Analysis

A preliminary view to the error distribution is given in the following graph, where errors are plotted as a function of insertion # for each test and each subject.

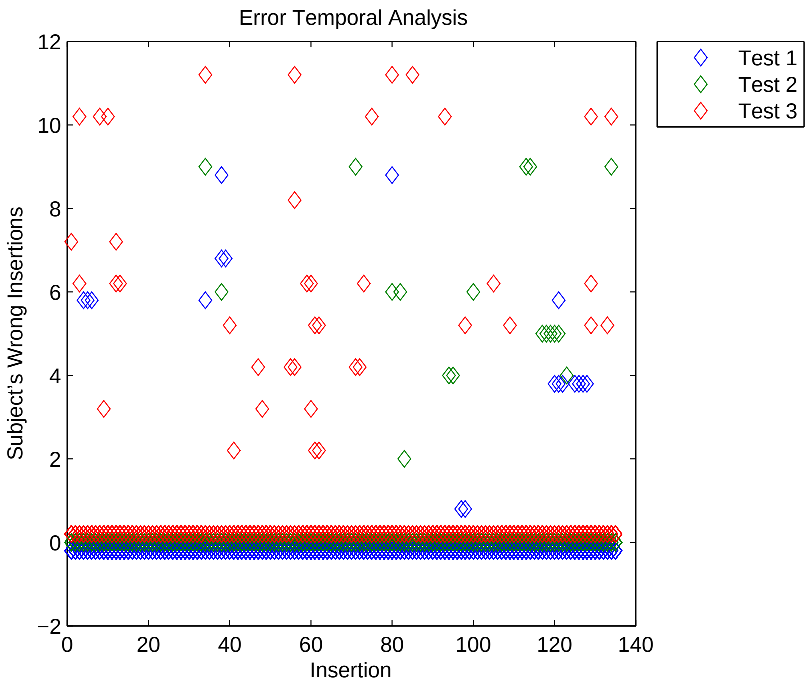


Figure 4.7: Error Distribution over Time per Subject

The following graph on the other hand shows how the number of errors were distributed when compared with the mean insertion time for each subject.

The following graphs show instead the error percentage in the three setups and the insetion time at which they occourred. Each column indicates the percentage of wrong insertions given the total number of insertions executed in a certain time.

As for the performance analysis, the Error analysis was executed in SAS and returned the following results:

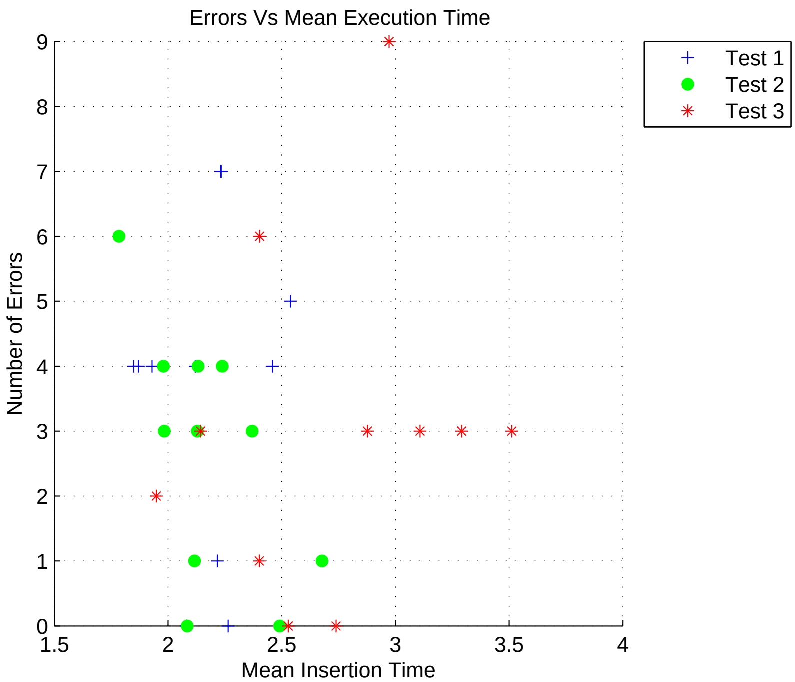


Figure 4.8: Errors made vs Mean insertion time

- SAS: ANOVA Mixed Model Analysis

SAS: ANOVA Mixed Model Analysis Results

Parameter	All Tests	Test 1	Test 2	Test 3
Cov. SubjectID	0.000176	0.000232	0.000154	0.000269
PinPerTask	0.2887	0.0244	0.7367	0.4765
Difficulty	0.0176	0.0008	0.2099	0.1587
PinPerTas*Difficulty	0.0565	0.0116	0.0129	0.3764
TestID	0.0099	–	–	–
PinPerTask*TestID	0.2415	–	–	–
Difficulty*TestID	0.0219	–	–	–
PinPer*Diffic*TestID	0.0127	–	–	–

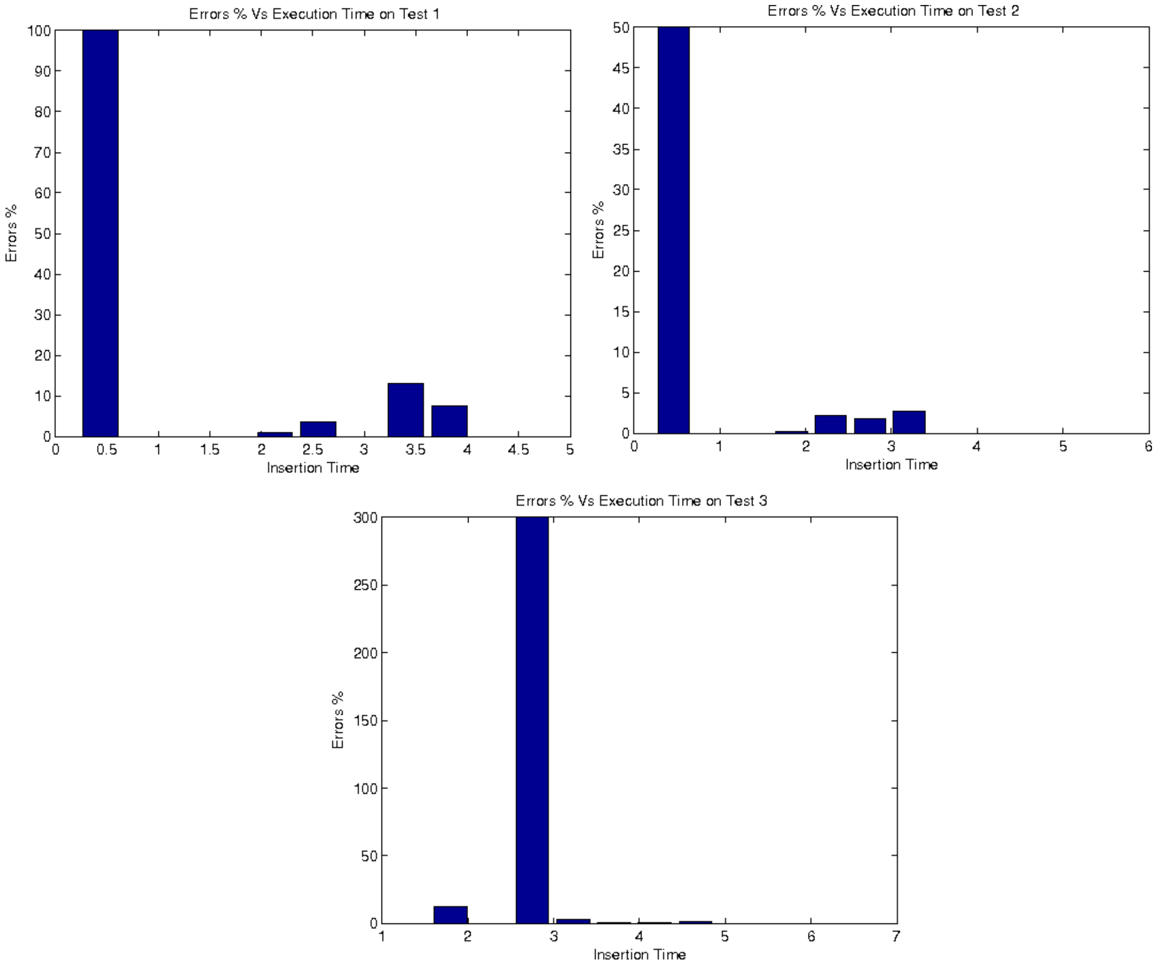


Figure 4.9: Error Distribution over Time per Subject

Parameter	Test 1-2	Test 2-3	Test 1-3
Cov. SubjectID	0.000154	0.000158	0.000163
PinPerTask	0.4811	0.3288	0.2091
Difficulty	0.0356	0.2119	0.0048
PinPerTas*Difficulty	0.0402	0.1284	0.1252
TestID	0.4870	0.0178	0.0669
PinPerTask*TestID	0.0528	0.8142	0.2212
Difficulty*TestID	0.0073	0.1316	0.2218
PinPer*Diffic*TestID	0.0092	0.2415	0.2675

The results above indicate that there is no real trend in the Error distribution and therefore it is safe to assume that errors are random. The results also indicate that subject's variance on errors was negligible, therefore consolidating the hypothesis of a statistically significant population. The only relevant trend that one might see was in the first test model where results seem to indicate that errors depended on difficulty. This hypothesis was denied by looking at the raw data and assessing that for test 1 only 15 errors were made. The very few samples, make the error analysis very unreliable, and therefore results might indicate false trends.

- SAS: ANOVA Logistic Regression Model Analysis

Now the next question would be, what is the probability of an error occurring in each test? In order to answer this question a Logistic Regression analysis was executed in SAS.

- Generic Logistic Regression Model SAS Input file:

```
filename resfile '/homes/maxdc/SAS/DataSAS';
    *Read from file;
    data subjects;
    infile resfile DLM=',' FIRSTOBS=2;
    input SubjectID PinPerTask Pin Hole Difficulty TestID Time Correct;
    run;
    *Create model;
    proc logistic data=subjects;
    class Correct PinPerTask Difficulty TestID SubjectID;
    model Correct = PinPerTask Difficulty TestID SubjectID ;

    run;
```

The following analysis evaluated the odds of a error occurring for each of the three tests:

Test	Error Probability (%)
Test 1	0.5 %
Test 2	0.8 %
Test 3	1.3 %

4.3.4 User Experience

User responses to the three setups can be summarized by the following tables:

Cards (Responce varies from 1 (disagree) to 5 (agree))

	Mean responce	Interpretation
Were the trials easy?	4.45	Trials were very easy
Were the cards hard to read?	1.82	Cards were easy to read
Did you have any difficulty distinguishing the pins?	1.36	Pins were easy to distinguish
Did you have any difficulty distinguishing the holes?	1.5	Holes were easy to distinguish
Were the feedback tones clear?	4.55	Tones were very clear
Were you satisfied with your performance?	4.18	Satisfied

From the above results we can see that in general subjects found the first test to be very simple and intuitive. Overall they were satisfied with their performance. From the experiment notes, it was interesting to note that all test subjects kept the deck of cards on their dominant hand side. All test subjects kept the deck on the table in front of them, except for one who kept it on his lap. It was interesting to see that 60 % of the subjects occasionally, before the insertion of the pin, verified the information on the card and then inserted the pin. This practice was not always adopted but it seemed to happen more often thowards the end of the trial. Other important notes were that in general subjects used only their dominant hand when executing the task. Only 3 subjects used two hands but not all the time. It was interesting to note also that subjects in general repeated outloud the instruction they were executing.

,

HMD (cards) (Responce varies from 1 (disagree) to 5 (agree))

	Mean response	Interpretation
Were the trials easy?	4.27	Trials were easy
Was the HMD hard to use?	2	HMD was easy to use
Did you have any difficulty distinguishing the pins?	1.73	Pins easy to distinguish
Did you have any difficulty distinguishing the holes?	1.55	Holes easy to distinguish
Was the HMD interface intuitive?	4.27	HMD Interface was intuitive
Did the HMD obstruct your view too much?	2.55	HMD somewhat obstructed FOV
Was the HMD comfortable?	2.83	HMD somewhat confortale
Did you experience any side effects?	2.36	Mild sideffects
Were sound warnings clear?	4.27	Tones were very clear
Were you satisfied with your performance?	4.18	Satisfied

, During this second trial, subjects found the task to be slightly harder than the previous. Form the responses, this could be due to the different focal distance between the virtual cards and the board. Some subjects found it very hard (or impossible) to focus at a virtual object in the visor, and decided to raise their head positioning the virtual image outside their main workspace. By raising their head, they saw that if no objects were located in close proximity they could focus on the visor image. Apparently they found it unnatural to focus behind a real object that was located directly in their main field of view. It is important to note also that in general mild sideffects were experienced and possibly they were due to the unnatural focusing exercise that was required. Generally sideffects faded once the HMD was removed. No test subject asked to remove the HMD during the test although they were asked several times.

HMD (Graphical Overlay) (Responce varies from 1 (disagree) to 5 (agree))

	Mean responce	Interpretation
Were the trials easy?	3	Trials were moderately hard
Was the HMD hard to use?	3.82	HMD somewhat hard to use
Did you have any difficulty distinguishing the pins?	2.91	Pins neither hard nor easy to distinguish
Did you have any difficulty distinguishing the holes?	2.91	Holes neither hard nor easy to distinguish
Was the HMD interface intuitive?	3.73	Interface not too intuitive
Did the HMD obstruct your view too much?	2.73	HMD slightly obstructed FOV
Was the HMD comfortable?	2.73	HMD not very comfortable
Did you experience any side effects?	2.82	Mild sideffects
Were sound warnings clear?	4.36	Tones were very clear
Were you satisfied with your performance?	3.09	Neutral

In the last trial as the user responce assesses, subjects found the task to be much more difficult. It is reasonable to assume that this was due to the HMD characteristics more than on the semantics behind the test. In this scenario the user was asked to cover one eye and this reduced their depth perception. Test subjects were noticeably slower also due to difficulties in finding, grasping and inserting properly the pins. There were also other aspects of the HMD setup which did not make it optimal. First, the HMD field of view did not allow the subject to see the entire board and second, the camera refresh rate didn't allow smooth projections of the graphical overlay. These factors plus the different focal distance between the graphic overlay and the board, probably caused the evident performance deterioration.

The final questionnaire also included two optional sections where test subjects could note their impressions and comments. Comments are summarized below.

Features that you would like to see in future HMD's:

- Fine adjustment of glasses
- Stereo vision
- Better camera mount for easy adjustment
- More confortable head mount
- Larger Field of View
- Image stabilization to reduce virtual image wobbling
- Faster resolution time
- Better registration
- Variable HMD focus distance
- Higher camera refresh rate
- Lighter system
- Less cables, possibly wireless system

General Comments:

- Consider bolting down the Pin-Hole Board
- Wireless pins would be better in the future
- HMD checklist with less jitter
- One subject preferred to look up into visor rather than have it overlay board (Test 2)
- One subject felt like he could perform additional mental tasks during test 3
- Subjects though that the board could be done better
- Holes happend to clog
- Cables happend to tangle

Chapter 5
CONCLUSIONS AND FUTURE WORK
5.1 Summary

Augmented Reality and most of all see-through HMD technology captured our attention in the first place due to their ability to give the user access to digital information rapidly, easily and in confined spaces such as the spacesuit environment. Future space missions will see astronauts on their own, facing complex challenges. This scenario somewhat resembles our life here on Earth where everyday we face new challenges that not necessarily we know how to tackle. This train of though lead to the following question: "If I was on Mars and I had to execute an experiment while on EVA or if I had to service a system, what would I ideally need or want?" The answer today would probably be, first of all the tools required for the task, and then a laptop with internet access so that if I don't know or I don't remember something, I could look it up. As funny as it can sound, computers and access to a database, have become the main source of information today. By then, the question was defined. How do we give astronauts the ability to access information from a computer while in a spacesuit?

The SSL at the University of Maryland was the perfect setup for answering this question, and the MX-2 space suit analogue an extremely useful tool. Modern spacesuits, are small, uncomfortable and do not give the astronaut too much hand mobility, therefore simply giving them a laptop would not work. In the specific case of the MX-2, a computer was already present in the suit's PLSS, therefore the challenge focused mainly on how to give access to it to the subject in the suit.

In the SSL research was made in terms of voice recognition, therefore the user interaction part was being studied. On the other hand, the only output to the subject was audio feedback, and it wasn't nearly sufficient. The fist attempt made was to introduce a PDA outside the suit that allowed the user to access video feedback, followed by an internal Head Up Display. Both systems gave the subject in the suit a better situational awareness and the ability to access digital information such as checklists and diagrams. This first system, was meant to be unintrusive and was positioned outside the user's main field of view. Although very useful, the

HUD was merely a substitute for a monitor. By this time bibliographical research pointed out that if we could introduce an in-field-of-view display system, we would have been able to not only do what we were achieving with the HUD, but also introduce AR technology. AR would give us the ability to recognize objects and display important information specific to the object of attention. This last feature gave birth to a new concept as: If we can recognize objects and display information within the users filed of view, we might also be able to relieve the user of some mental workload. The HMD was the first attempt toward this concept. It was soon seen that see-through HMDs were not the most wide spread and affordable systems, therefore due to availability an old and rahter low performance HMD was found and used in order to learn more on these systems. The poor performance of the HMD rapidly pointed out a new set of questions. Undertanding and defining the fundamental characteristics that would make HMD systems first of all, useful, then efficient was paramount. Testing was necessary and the experiment basic concept was born.

5.2 Conclusions

Many were the question we asked ourselves in this work. All of them found and answer but most of the time answers lead to more questions. The first conclusion drawn was that this work is by no means conclusive, and a lot of effort is still necessary in order to deliver a functional and efficient AR HMD. All the phases in this reseach were very useful in defining the problem in increasing detail and led to understanding the fundamental parameters that could make an HMD system usable. It is not possible to say at this point in time if the system would be more efficient than current systems, but this hypothesis is not denied either.

The previous experimental data analysis could mislead the reader by assessing that the HMD system didn't deliver the expected performance, on the other side, what should be leared from the results is that the HMD was not ready yet to be compared with simpler systems. It is important to note that the experiment taught us many things. First of all, it was possible to execute a simple task without any prior knowledge and without asking the user to read or understand anything. This was not happly accepted by the test subjects since it gave them an uncomfortable feeling of not being in control of what they were asked to perform. This result tells us that mental workload and user acceptance is not a linear function, and there is a point where if one reduces the mental workload too much, a task no matter how simple, could become uncomfortable to execute and prone to errors. During the HMD graphical overlay testing it was often seen that subjects tried to complicate the task by trying to gather more information. Subjects counted the displayed holes and pins and executed the derived instruction instead of just following the displayed line.

Another important aspect of the experimental setup to keep in mind was that the HMD delivered in general poor performance due to its intrinsic characteristics. The HMD used is not able to display stereoscopic images at different focal distances or separations, therefore the user had to continuously change focus depth to receive the task information and they were forced to cover one eye and lose depth perception to eliminate ambiguities in the displayed information (Test 3). Also, the HMD field of view did not allow the user to see the entire board and required the subjects to continuously move their head. This pointed out another problem with the setup which was the camera refresh rate. The camera and the computer were not nearly as fast as the human eyes and mind in re-recognizing an object and displaying the new virtual image. The lag between the real and the synthetic worlds caused the subjects to have to wait for the system to stabilize and this might have been one of the reasons for a general overhead in the performance acquired. Lastly as it was seen previously sideffects could not be ignored, but they could be reasonably reconducted to the previously described flaws in the HMD. It is very possible that by using a more efficient HMD, these sideffects, could at least be mitigated, if not eliminated.

In conclusion the third experimental setup was not comparable in difficulty to the first two and some major modifications to the HMD systems will have to be made in order to repeat the experiment and compare the results.

In regard to the first two scenarios, results showed that there is no significant difference between the two setups. This result denied for the time beeing the initial hypothesis therefore we cannot assess any particular benefit in performance or accuracy due to the use of in-field-of-view information displays.

This work allowed us to understand where to focus our attention, and assessed that there is a very likely possibility of achieving a more efficient HMD system in the near future.

5.3 Future Reseach

5.3.1 Hardware Development

Hardware development of the HMD system will be fundamental for future experimentation and should mainly focus on wider virtual displays that would allow a more natural coexistence of the virtual and real worlds. Higher resolution and contrast would increase the amount of information that could be displayed as well as increasing the coexistence illusion. Very important will be the introduction of stereoscopic capabilities and variable focus distance. This last features will possibly reduce or eliminate sideffects and might increase the user performance dramatically. It might be intersting to increase also the camera's field of view and resolution in order to improve target recognition but more important would be the camera refresh rate. On this point it might be usefull to equip the system with an inertial measurement unit which bandwidth would reduce the effects of the slow refresh rate of the cameras. In this case the system would rely on the IMU for fast motion, and it would recalibrate and register the virtual image through the cameras, which at that point would not require a higher refresh rate. Lastly, the HMD and camera mounting should be improved and should allow easier individual configuration of the system as well as improve confort.

5.3.2 Software Development

Software development will play a major role in the future implementations of the system. Many are the features that could be implemented, a few near term objectives could be 3D interactive maps, video display and database acess. It will be important also to improve the pattern recognition algorithm in order to minimize the image processing time, and accuracy. It will also be fundamental to integrate the AR software with the voice recognition suite. It might also be intersting to introduce asynchronous stereo cameras on the HMD in order to increase the depth perception of the system as well as its refresh rate. More research should also be devoted in defining display modes that avoid cluttering of the visible workspace and that increase the subject's situational awareness.

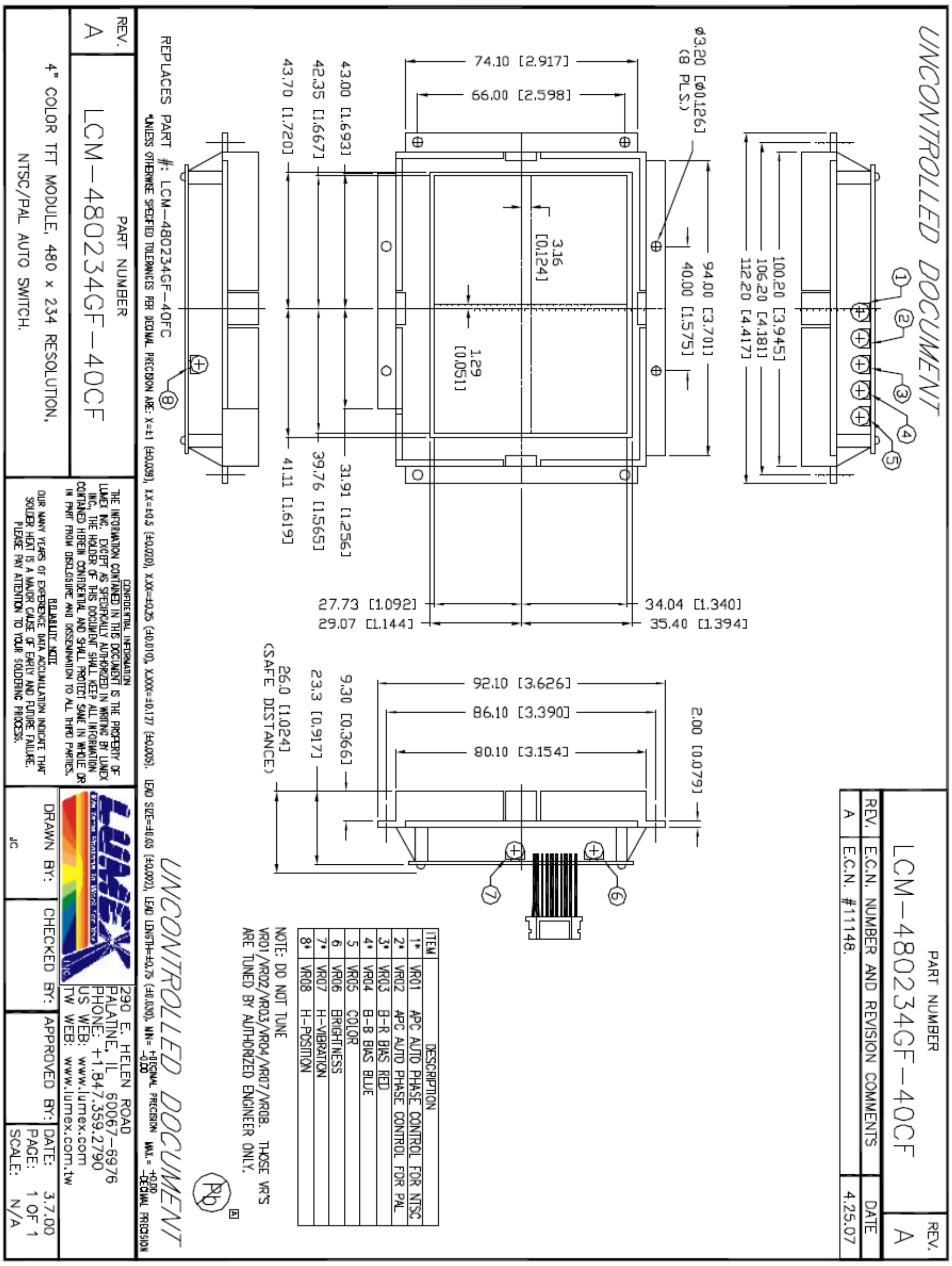
5.3.3 Experimental Research

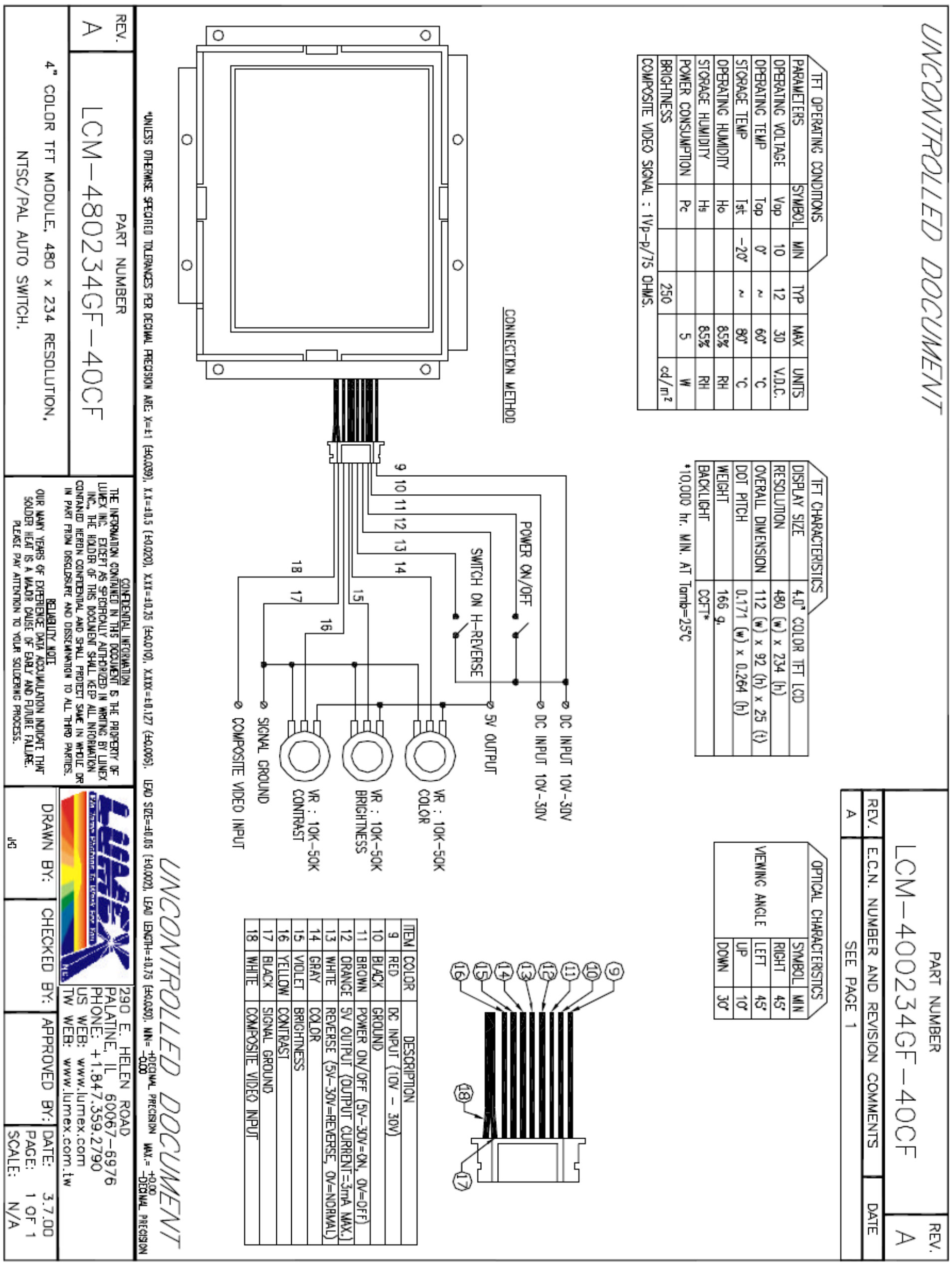
Experimentation should be oriented mainly in the assessment of the user experience and their performance with the new HMD system. In the previous experiment it was noted that the proposed task was possibly too simple, and it did not return a lot of information regarding higher complexity tasks. It might be interesting to repeat the previous experiment with a much more complex task and see how subjects respond. It is reasonable that the trends found in the previous analysis could be very different if the task difficulty is increased, especially if the mean execution time allowed the HMD overhead in performance to be negligible when compared to the overall execution time. A possible experimental setup could be to ask the subjects to disassemble, identify a fault and reassemble a system while receiving information from a manual and from and HMD with graphical overlay. As long as the system is compatible in dimensions with the HMD field of view, it might be very interesting to see how performance and accuracy in the execution varies. The previous experiment results could also be seen as an upped boundary in performance and accuracy, therefore it will be interesting to study the ideal lower boundary. In order to do so, Video interfaces could be used in order to mitigate the sideffects of optical interfaces. These would allow us eliminate the focal distance and stereoscopy problem. In order to do so, a possible setup would see the subject's executing a simple task displayed on a computer monitor. The task in particular could be very similar to the pin-hole task seen before, with the only difference that the entire setup would be virtual. This would allow us to acquire accuracy, performance, as well as movement trajectories, etc. In conclusion, although performance is a fundamental parameter, for the specific application we are considering, more attention should be given in finding a display interface that increases accuracy more than anthing else. For this purpose future system developpment could include the integration of automatic error checking. This last point would open new study areas such as how to check for errors and display the adeguate information.

Appendix A

HARDWARE SPECIFICATIONS

A.1 LUMIX LCM 480234GF-40CF DATA SHEET





A.2 SONY Glasstron PLM-S600

Features:

- Video Playback From any source with phono connections. Plug into a VCR, DVD Player, Video Walkman or Camcorder using standard phono connections;
- 52" Virtual Viewing Glasstron reproduces the feel of viewing a 52" televisoin at 6-1/2 feet; you'll think you're watching a large screen system;

- HiFi Stereo Sound with 3 Modes Personal headphones deliver full stereo sound and with surround sould you feel like you're right in the action of your favorite movie. The AVLS (Auto Volume Limiter System) limits the sound that escapes from the headphones, preventing you from disturbing others in the earea;

- Adjustable Head Support System Using two adjustable straps you can customize the fit of your Glasstron. Adjust the back strap for a snug front to back feel and the suspender head piece for proper viewing. A step by step menu system helps insure proper LCD screen alignment;

- Various Viewing Modes -See Through Mode: Allows for a variable degree of viewing your picture, you can adjust how much of your outside environment you let in or close out;

- Screen Mode: Places your picture in the middle of the screen and surrounds the picture with the outside environment;

- A/V Mute: Turns off the sound and picture, and lets in the outside environment;

- Color / Hue / Brightness Control;
- Indoor Outdor Use A custom three digit password is available for child protection (Glasstron PLM-100 is not recommended for children age 15 or younger).
- Resolution: 648 x 486 (NTSC);
- Transparency: 30/100;

Supplied Accessories:

- A/V Cable (mini-to-mini plug);
- A/VCable (mini-to-mini RCA pin);
- A/C Adapter;
- Soft Carry Case;

Optional Accessories:

- Lithium Ion Battery NP-F550, NP-F750, NP-F950

Weight:

- Headset: Approx. 12 oz;
- Control Unit: Approx. 4 oz .

Dimensions:

- 10 1/4" X 4 3/4" X 10 1/4"

(Source: http://www.acadia.org/competition-98/sites/integrus.com/html/library/tech/www.sel.sony.com/SEL/consumer/ss5/home/camcorder/camcorders8mmaccessories/ plm-100_specs.html)

A.3 Apple iSight Camera

Manufacturer: Apple Inc.

Part number: M8817LL/C

Device Type: Web camera

Optical sensor size : 1/4 in

Weight: 2.3 oz

Optical sensor type: CCD

Type: Color

Audio Support: Yes

Lens Aperture: F/2.8

Video input features: Digital noise reduction

Focus Adjustment: Automatic

Min Focus Range: 2 in

Interfaces: IEEE 1394 (6 pin FireWire)

A.4 GENERAL ELECTRIC EasyCam

Key Features:

- Interface Type: USB
- Video Capture Resolution: 640 x 480
- Digital Video Capture Speed: 30 frames per second
- Still Image Capture Resolution: 640 x 480

Other Features:

- Digital Zoom: 2x
- Color Depth: 24 Bit

A.5 NATIONAL INSTRUMENTS USB-6008

Low-Cost Multifunction DAQ for USB

NI USB-6008, NI USB-6009

- Small and portable
- 12 or 14-bit input resolution, at up to 48 kS/s
- Built-in, removable connectors for easier and more cost-effective connectivity
- 2 true DAC analog outputs for accurate output signals
- 12 digital I/O lines (TTL/LVTTL/CMOS)
- 32-bit event counter
- Student kits available
- OEM versions available

Operating Systems
- Windows 2000/XP
- Mac OS X[1]
- Linux®[1]
- Pocket PC
- Win CE

Recommended Software
- LabVIEW
- LabWindows/CVI

Measurement Services Software (included)
- NI-DAQmx
- Ready-to-run data logger

[1]Mac OS X and Linux users need to download NI-DAQmx Base.

Product	Bus	Analog Inputs[1]	Input Resolution (bits)	Max Sampling Rate (kS/s)	Input Range (V)	Analog Outputs	Output Resolution (bits)	Output Rate (Hz)	Output Range (V)	Digital I/O Lines	32-Bit Counter	Trigger
USB-6009	USB	8 SE/4 DI	14	48	±1 to ±20	2	12	150	0 to 5	12	1	Digital
USB-6008	USB	8 SE/4 DI	12	10	±1 to ±20	2	12	150	0 to 5	12	1	Digital

[1]SE = single ended, DI = differential

Hardware Description

The National Instruments USB-6008 and USB-6009 multifunction data acquisition (DAQ) modules provide reliable data acquisition at a low price. With plug-and-play USB connectivity, these modules are simple enough for quick measurements but versatile enough for more complex measurement applications.

Software Description

The NI USB-6008 and USB-6009 use NI-DAQmx high-performance, multithreaded driver software for interactive configuration and data acquisition on Windows OSs. All NI data acquisition devices shipped with NI-DAQmx also include VI Logger Lite, a configuration-based data-logging software package.

Mac OS X and Linux users can download NI-DAQmx Base, a multiplatform driver with a limited NI-DAQmx programming interface. You can use NI-DAQmx Base to develop customized data acquisition applications with National Instruments LabVIEW or C-based development environments. NI-DAQmx Base includes a ready-to-run data logger application that acquires and logs up to eight channels of analog data.

PDA users can download NI-DAQmx Base for Pocket PC and Win CE to develop customized handheld data acquisition applications.

Recommended Accessories

The USB-6008 and USB-6009 have removable screw terminals for easy signal connectivity. For extra flexibility when handling multiple wiring configurations, NI offers the USB-6008/09 Accessory Kit, which includes two extra sets of screw terminals, extra labels, and a screwdriver.

In addition, the USB-6008/09 Prototyping Accessory provides space for adding more circuitry to the inputs of the USB-6008 or USB-6009.

Common Applications

The USB-6008 and USB-6009 are ideal for a number of applications where economy, small size, and simplicity are essential, such as:
- Data logging – Log environmental or voltage data quickly and easily.
- Academic lab use – The low price facilitates student ownership of DAQ hardware for completely interactive lab-based courses. (Academic pricing available. Visit **ni.com/academic** for details.)
- Embedded OEM applications.

NATIONAL INSTRUMENTS

Information for Student Ownership

To supplement simulation, measurement, and automation theory courses with practical experiments, NI has developed the USB-6008 and USB-6009 student kits, which include the LabVIEW Student Edition and a ready-to-run data logger application. These kits are exclusively for students, giving them a powerful, low-cost hands-on learning tool. Visit **ni.com/academic** for more details.

Information for OEM Customers

For information on special configurations and pricing, call (800) 813 3693 (U.S. only) or visit **ni.com/oem**. Go to the Ordering Information section for part numbers.

Ordering Information

NI USB-6008[1]	779051-01
NI USB-6009[1]	779026-01
NI USB-6008 OEM	193132-02
NI USB-6009 OEM	193132-01
NI USB-6008 Student Kit[1,2]	779320-22
NI USB-6009 Student Kit[1,2]	779321-22

[1] Includes NI-DAQmx software, NI ready-to-run data logger software, and a USB cable.

[2] Includes LabVIEW Student Edition.

BUY NOW!

For complete product specifications, pricing, and accessory information, call 800 265 9891 (U.S. only) or go to ni.com/usb.

Specifications

Typical at 25 °C unless otherwise noted.

Analog Input

Absolute accuracy, single-ended

Range	Typical at 25 °C (mV)	Maximum (0 to 55 °C) (mV)
±10	14.7	138

Absolute accuracy at full scale, differential[1]

Range	Typical at 25 °C (mV)	Maximum (0 to 55 °C) (mV)
±20	14.7	138
±10	7.73	84.8
±5	4.28	58.4
±4	3.59	53.1
±2.5	2.56	45.1
±2	2.21	42.5
±1.25	1.70	38.9
±1	1.53	37.5

Number of channels .... 8 single-ended/4 differential
Type of ADC .... Successive approximation

ADC resolution (bits)

Module	Differential	Single-Ended
USB-6008	12	11
USB-6009	14	13

Maximum sampling rate (system dependent)

Module	Maximum Sampling Rate (kS/s)
USB-6008	10
USB-6009	48

Input range, single-ended .... ±10 V
Input range, differential .... ±20, ±10, ±5, ±4, ±2.5, ±2, ±1.25, ±1 V
Maximum working voltage .... ±10 V
Overvoltage protection .... ±35 V
FIFO buffer size .... 512 B
Timing resolution .... 41.67 ns (24 MHz timebase)
Timing accuracy .... 100 ppm of actual sample rate
Input impedance .... 144 k
Trigger source .... Software or external digital trigger
System noise .... 0.3 LSB_{rms} (±10 V range)

Analog Output

Absolute accuracy (no load) .... 7 mV typical, 36.4 mV maximum at full scale
Number of channels 2
Type of DAC .... Successive approximation
DAC resolution .... 12 bits
Maximum update rate .... 150 Hz, software-timed
Output range .... 0 to +5 V
Output impedance .... 50 Ω
Output current drive .... 5 mA
Power-on state .... 0 V
Slew rate .... 1 V/μs
Short-circuit current .... 50 mA

Digital I/O

Number of channels .... 12 total
8 (P0.<0..7>)
4 (P1.<0..3>)
Direction control .... Each channel individually programmable as input or output
Output driver type
USB-6008 .... Open-drain
USB-6009 .... Each channel individually programmable as push-pull or open-drain
Compatibility .... CMOS, TTL, LVTTL
Internal pull-up resistor .... 4.7 kΩ to +5 V
Power-on state .... Input (high impedance)
Absolute maximum voltage range .... -0.5 to +5.8 V

Digital logic levels

Level	Min	Max	Units
Input low voltage	-0.3	0.8	V
Input high voltage	2.0	5.8	V
Input leakage current	–	50	μA
Output low voltage (I = 8.5 mA)	–	0.8	V
Output high voltage (push-pull, I = -8.5 mA)	2.0	3.5	V
Output high voltage (open-drain, I = -0.6 mA, nominal)	2.0	5.0	V
Output high voltage (open-drain, I = -8.5 mA, with external pull-up resistor)	2.0	–	V

Counter

Number of counters 1
Resolution .... 32 bits
Counter measurements .... Edge counting (falling edge)
Pull-up resistor .... 4.7 kΩ to 5 V
Maximum input frequency .... 5 MHz
Minimum high pulse width .... 100 ns
Minimum low pulse width .... 100 ns
Input high voltage .... 2.0 V
Input low voltage .... 0.8 V

Power available at I/O connector

+5 V output (200 mA maximum) .... +5 V typical
+4.85 V minimum
+2.5 V output (1 mA maximum) .... +2.5 V typical
+2.5 V output accuracy .... 0.25% max
Voltage reference temperature drift .... 50 ppm/°C max

[1]Input voltages may not exceed the working voltage range.

Physical Characteristics

If you need to clean the module, wipe it with a dry towel.

Dimensions (without connectors)	6.35 by 8.51 by 2.31 cm (2.50 by 3.35 by 0.91 in.)
Dimensions (with connectors)	8.18 by 8.51 by 2.31 cm (3.22 by 3.35 by 0.91 in.)
Weight (without connectors)	59 g (2.1 oz)
Weight (with connectors)	84 g (3 oz)
I/O connectors	USB series B receptacle (2) 16-position (screw-terminal) plug headers
Screw-terminal wiring	16 to 28 AWG
Screw-terminal torque	0.22 to 0.25 N•m (2.0 to 2.2 lb•in.)

Power Requirement

USB (4.10 to 5.25 VDC)	80 mA typical 500 mA maximum
USB suspend	300 µA typical 500 µA maximum

Environmental

The USB-6008 and USB-6009 are intended for indoor use only.

Operating environment	
Ambient temperature range	0 to 55 °C (tested in accordance with IEC-60068-2-1 and IEC-60068-2-2)
Relative humidity range	10 to 90%, noncondensing (tested in accordance with IEC-60068-2-56)
Storage environment	
Ambient temperature range	-40 to 85 °C (tested in accordance with IEC-60068-2-1 and IEC-60068-2-2)
Relative humidity range	5 to 90%, noncondensing (tested in accordance with IEC-60068-2-56)
Maximum altitude	2,000 m (at 25 °C ambient temperature)
Pollution degree	2

Safety and Compliance

Safety

This product is designed to meet the requirements of the following standards of safety for electrical equipment for measurement, control, and laboratory use:

- IEC 61010-1, EN 61010-1
- UL 61010-1, CAN/CSA-C22.2 No. 61010-1

Note: For UL and other safety certifications, refer to the product label or visit **ni.com/certification**, search by model number or product line, and click the appropriate link in the Certification column.

Electromagnetic Compatibility

This product is designed to meet the requirements of the following standards of EMC for electrical equipment for measurement, control, and laboratory use:

- EN 61326 EMC requirements; Minimum Immunity
- EN 55011 Emissions; Group 1, Class A
- CE, C-Tick, ICES, and FCC Part 15 Emissions; Class A

Note: For EMC compliance, operate this device according to product documentation.

CE Compliance

This product meets the essential requirements of applicable European Directives, as amended for CE marking, as follows:

- 73/23/EEC; Low-Voltage Directive (safety)
- 89/336/EEC; Electromagnetic Compatibility Directive (EMC)

Note: Refer to the Declaration of Conformity (DoC) for this product for any additional regulatory compliance information. To obtain the DoC for this product, visit **ni.com/certification**, search by model number or product line, and click the appropriate link in the Certification column.

Waste Electrical and Electronic Equipment (WEEE)

EU Customers: At the end of their life cycle, all products must be sent to a WEEE recycling center. For more information about WEEE recycling centers and National Instruments WEEE initiatives, visit **ni.com/environment/weee.htm**.

NI Services and Support

NI has the services and support to meet your needs around the globe and through the application life cycle – from planning and development through deployment and ongoing maintenance. We offer services and service levels to meet customer requirements in research, design, validation, and manufacturing. Visit **ni.com/services**.

Training and Certification

NI training is the fastest, most certain route to productivity with our products. NI training can shorten your learning curve, save development time, and reduce maintenance costs over the application life cycle. We schedule instructor-led courses in cities worldwide, or we can hold a course at your facility. We also offer a professional certification program that identifies individuals who have high levels of skill and knowledge on using NI products. Visit **ni.com/training**.

Professional Services

Our Professional Services Team is comprised of NI applications engineers, NI Consulting Services, and a worldwide National Instruments Alliance Partner program of more than 600 independent consultants and integrators. Services range from start-up assistance to turnkey system integration. Visit **ni.com/alliance**.

OEM Support

We offer design-in consulting and product integration assistance if you want to use our products for OEM applications. For information about special pricing and services for OEM customers, visit **ni.com/oem**.

Local Sales and Technical Support

In offices worldwide, our staff is local to the country, giving you access to engineers who speak your language. NI delivers industry-leading technical support through online knowledge bases, our applications engineers, and access to 14,000 measurement and automation professionals within NI Developer Exchange forums. Find immediate answers to your questions at **ni.com/support**.

We also offer service programs that provide automatic upgrades to your application development environment and higher levels of technical support. Visit **ni.com/ssp**.

Hardware Services

NI Factory Installation Services

NI Factory Installation Services (FIS) is the fastest and easiest way to use your PXI or PXI/SCXI combination systems right out of the box. Trained NI technicians install the software and hardware and configure the system to your specifications. NI extends the standard warranty by one year on hardware components (controllers, chassis, modules) purchased with FIS. To use FIS, simply configure your system online with **ni.com/pxiadvisor**.

Calibration Services

NI recognizes the need to maintain properly calibrated devices for high-accuracy measurements. We provide manual calibration procedures, services to recalibrate your products, and automated calibration software specifically designed for use by metrology laboratories. Visit **ni.com/calibration**.

Repair and Extended Warranty

NI provides complete repair services for our products. Express repair and advance replacement services are also available. We offer extended warranties to help you meet project life-cycle requirements. Visit **ni.com/services**.

ni.com • (800) 813 3693
National Instruments • info@ni.com

351378A-01 2006-7322-301-101-D

Appendix B

CODE

B.1 Experiment Software: DAQacq.cpp

This program, is responsible for acquiring data from the two NI-DAQ 6008 cards. It identifies pin insertions, and compares them with the reference file. It also produces a log file that includes pin insertion times and whether the insertion was wrong or right. the program was written in C++ and uses specific NI libraries and drivers. This version is only Mac OS-X compatible.

B.1.1 Code:

```
#include <iostream>
#include <iomanip>
#include "NIDAQmxBase.h"
#include <stdio.h>
#include <time.h>
#include <fstream>
#include <sstream>
#include <string>

#define DAQmxErrChk(functionCall) { if( DAQmxFailed(error=(functionCall)) ) { goto Error; } }

using namespace std;

int main(int argc, char *argv[])
{
    // Task parameters
    int32       error = 0;
    TaskHandle  taskHandle1 = 0;
    TaskHandle  taskHandle2 = 0;
    char        errBuff[2048]={'\0'};

    // Channel parameters
    char        chan1[] = "Dev1/ai0, Dev1/ai1, Dev1/ai2, Dev1/ai3";
    char        chan2[] = "Dev2/ai0, Dev2/ai1, Dev2/ai2";
    char        FileName[256];
    char        TaskFile[256];
    string      line;
    int         int line;
    float64     min = -5.0;
    float64     max = 5.0;

    // Timing parameters
    uInt64      samplesPerChan1 = 4;
    uInt64      samplesPerChan2 = 3;

    // Data read parameters
    float64     data1[4];
    float64     data2[3];
    int         datap[7]={1, 2, 3, 4, 5, 6, 7};
    int32       pointsToRead1 = 4;
    int32       pointsToRead2 = 3;
    int32       pointsRead;
    float64     timeout = 10.0;

    int         npins;
    std::string trash;
    timeval start, end;
    double tmp1, tmp2, tottime, AVGtime;
    int dataStored1[]={0, 0, 0, 0};
    int dataStored2[]={0, 0, 0};
    int tottrials;
    int kk, jj;

    //Set Logfile
    printf ("\n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n" );
    cout << "Enter Test Subject Id #:  (es. 1234): ";
    cin >> FileName ;
    cout << "Enter Task File Name:  (es. A): ";
    cin >> TaskFile ;

    ifstream myfile2 (TaskFile);
    if (myfile2.is_open()) {printf("Task File Opened \n");
    } else {
        printf("ERROR: Could not open Task File \n");
        return 0;
    }

    getline (myfile2,line);
    istringstream buffer(line);
    buffer >> tottrials;

    ofstream myfile;
    myfile.open(FileName);
    myfile <<"              Subject              "<< (FileName) <<" \n";
    myfile <<"              Task File        "<< TaskFile <<" \n"  ;
    myfile <<"              Total Number of Trials :  "<< tottrials <<" \n"  ;

    //Set NIDAQ cards
    printf ("Initializing... \n ");
    DAQmxErrChk (DAQmxBaseCreateTask("",&taskHandle1));
    DAQmxErrChk (DAQmxBaseCreateAIVoltageChan(taskHandle1,chan1,"",DAQmx_Val_Cfg_Default,min,max,DAQmx_Val_Volts,NULL));
    DAQmxErrChk (DAQmxBaseStartTask(taskHandle1));
    DAQmxErrChk (DAQmxBaseCreateTask("",&taskHandle2));
    DAQmxErrChk (DAQmxBaseCreateAIVoltageChan(taskHandle2,chan2,"",DAQmx_Val_Cfg_Default,min,max,DAQmx_Val_Volts,NULL));
    DAQmxErrChk (DAQmxBaseStartTask(taskHandle2));

    for (int ii=1;ii<=tottrials; ii++)
    {
        myfile <<"              Trial              "<< ii <<" \n"  ;

        if (ii==1) {
            printf ("\n Trial %u \n",ii);
            //cout << "Enter # of pins to move in this trial: ";
        }else{
            cout << "\n ------------ RESET PINS ------------- \n";
            printf ("\n Trial %u \n",ii);
            //cout << "Enter # of pins to move in this trial: ";
        }
        getline (myfile2,line);
        istringstream buffer(line);
```

```
buffer >> npins;

myfile <<"              Pin Movements in Trial :  " << npins << " \n"  ;
kk=0;
tottime=0;
AVGtime=0;
dataStored1[0]=0;
dataStored1[1]=0;
dataStored1[2]=0;
dataStored1[3]=0;
dataStored2[0]=0;
dataStored2[1]=0;
dataStored2[2]=0;

printf("Press RETURN to START \n");
trash="trash";
if(ii==1){
std::getline(std::cin, trash);
std::getline(std::cin, trash);}
else std::getline(std::cin, trash);
gettimeofday(&start, NULL);
printf("\a");
printf("\n START ! \n");
for (;;) {

    DAQmxErrChk (DAQmxBaseReadAnalogF64(taskHandle1,pointsToRead1,timeout,DAQmx_Val_GroupByChannel,data1,
samplesPerChan1,&pointsRead,NULL));
    DAQmxErrChk (DAQmxBaseReadAnalogF64(taskHandle2,pointsToRead2,timeout,DAQmx_Val_GroupByChannel,data2,
samplesPerChan2,&pointsRead,NULL));

    for (int i=0; i<=3; i++){
        if (((data1[i] > 0.3) || (data1[i]<-0.3) ) && (dataStored1[i]==0))
        {
            printf("\a");
            gettimeofday(&end, NULL);
            tmp1 = start.tv_sec + (start.tv_usec / 1000000.0);
            tmp2 = end.tv_sec   + (end.tv_usec   / 1000000.0);
            double time1 = tmp2 - tmp1;
            if (data1[i]>-1.5 && data1[i]<-1.3) jj=0;
            if (data1[i]>-0.7 && data1[i]<-0.5) jj=1;
            if (data1[i]>0.5 && data1[i]<0.7) jj=2;
            if (data1[i]>1.1 && data1[i]<1.3) jj=3;
            if (data1[i]>1.7 && data1[i]<1.9) jj=4;
            if (data1[i]>2.5 && data1[i]<2.7) jj=5;
            if (data1[i]>3.5 && data1[i]<3.7) jj=6;

            getline (myfile2,line);
            istringstream buffer(line);
            buffer >> int_line;

            if ((int_line/10 == datap[jj]) && ((int_line-(int_line/10)*10)==7-i)) {
            myfile << setprecision(4) << (time1) <<" " << (datap[jj]) <<"  " << (7-i) <<" Correct \n"  ;
            }else{
            myfile << setprecision(4) << (time1) <<" " << (datap[jj]) <<"  " << (7-i) <<" Wrong \n"  ;
            }

            dataStored1[i]=1;
            kk++;
            tottime=time1+tottime;
            AVGtime=time1/npins+AVGtime;
            gettimeofday(&start, NULL);

            printf ("Pin Insertion Acquired %u / %u \n",kk,npins);
        }
    }
    for (int i=0; i<=2; i++){
        if (((data2[i] > 0.3) || (data2[i]<-0.3) ) && (dataStored2[i]==0))
        {
            printf("\a");
            gettimeofday(&end, NULL);
            tmp1 = start.tv_sec + (start.tv_usec / 1000000.0);
            tmp2 = end.tv_sec   + (end.tv_usec   / 1000000.0);
            double time1 = tmp2 - tmp1;
            if (data2[i]>-1.5 && data2[i]<-1.3) jj=0;
            if (data2[i]>-0.7 && data2[i]<-0.5) jj=1;
            if (data2[i]>0.5 && data2[i]<0.7) jj=2;
            if (data2[i]>1.1 && data2[i]<1.3) jj=3;
            if (data2[i]>1.7 && data2[i]<1.9) jj=4;
            if (data2[i]>2.5 && data2[i]<2.7) jj=5;
            if (data2[i]>3.5 && data2[i]<3.7) jj=6;

            getline (myfile2,line);
            istringstream buffer(line);
            buffer >> int_line;

            if ((int_line/10 == datap[jj]) && ((int_line-(int_line/10)*10)==3-i)) {
            myfile << setprecision(4) << (time1) <<" " << (datap[jj]) <<"  " << (3-i) <<" Correct \n"  ;
            }else{
            myfile << setprecision(4) << (time1) <<" " << (datap[jj]) <<"  " << (3-i) <<" Wrong \n"  ;
            }

            dataStored2[i]=1;
            kk++;
            tottime=time1+tottime;
            AVGtime=time1/npins+AVGtime;
            gettimeofday(&start, NULL);

            printf ("Pin Insertion Acquired %u / %u \n",kk,npins);
        }
    }
```

```
            if (kk==npins) {
                myfile <<"            Total Trial Time :            "<<(tottime) <<"\n";
                myfile <<"            Average Insertion Time :         "<<(AVGtime) <<"\n";


                printf("\a");
                printf("\a");
                printf("\a");
                break;
            }
        }

    }
    Error:
                if( DAQmxFailed(error) )
                    DAQmxBaseGetExtendedErrorInfo(errBuff,2048);
                if( taskHandle1!=0 ) {
                    DAQmxBaseStopTask(taskHandle1);
                    DAQmxBaseClearTask(taskHandle1);
                }
                if( taskHandle2!=0 ) {
                    DAQmxBaseStopTask(taskHandle2);
                    DAQmxBaseClearTask(taskHandle2);
                }

            if( DAQmxFailed(error) )
            printf("DAQmxBase Error: %s\n",errBuff);

            printf ("Test Completed ! \n");
            myfile.close();
            myfile2.close();
            return 0;

}
```

B.2 Experiment Software: Multi.cpp

This application is an extension of DAQacq. It adds to all the previously implemented features, the ability to recognize markers and overlay on the pin board simple graphics. This application uses OpenGL functions to draw circles over the pins and holes, and connects them with a thick white line. The position of the line varies depending on which pin is to be inserted in which hole. As in the previous application, the program acquires the necessary information through a reference file. The marker recognition is done in such a way, that as long as the camera can see one of the three markers, it will know where to register the overlay. This application is MAC OS-X specific, and uses the ARtoolKit and NI libraries as well as NI specific drivers.

B.2.1 Code:

```
#ifdef _WIN32
#include <windows.h>
#endif
#include <stdio.h>
#include <stdlib.h>
#ifndef __APPLE__
#include <GL/gl.h>
#include <GL/glut.h>
#else
#include <OpenGL/gl.h>
#include <GLUT/glut.h>
#endif
#include <AR/gsub.h>
#include <AR/video.h>
#include <AR/param.h>
#include <AR/ar.h>
#include <AR/arMulti.h>

#include <iostream>
#include <iomanip>
#include "NIDAQmxBase.h"
#include <time.h>
#include <fstream>
#include <sstream>
#include <string>

#define DAQmxErrChk(functionCall) { if( DAQmxFailed(error=(functionCall)) ) { goto Error; } }

using namespace std;

/* set up the video format globals */

#ifdef _WIN32
char            *vconf = "Data\\WDM_camera_flipV.xml";
#else
char            *vconf = "";
#endif

int             xsize, ysize;
int             thresh = 70;


char            *cparam_name    = "Data/camera_para.dat";
ARParam         cparam;

char                *config_name = "Data/multi/marker.dat";
ARMultiMarkerInfoT  *config;

static void   init(void);
static void   cleanup(void);
static void   keyEvent( unsigned char key, int x, int y);
static void   mainLoop(void);
static void   draw( double trans1[3][4], double trans2[3][4], int mode );
//------------DAQ-----------------------------------
// Task parameters
    int32       error = 0;

    char        errBuff[2048]={'\0'};

    // Channel parameters
    char        chan1[] = "Dev1/ai0, Dev1/ai1, Dev1/ai2, Dev1/ai3";
    char        chan2[] = "Dev2/ai0, Dev2/ai1, Dev2/ai2";
    char        FileName[256];
    char        TaskFile[256];
    string      line;
    int         int_line;


    // Timing parameters
    uInt64      samplesPerChan1 = 4;
    uInt64      samplesPerChan2 = 3;

    // Data read parameters
    float64     data1[4];
    float64     data2[3];
    int         datap[7]={1, 2, 3, 4, 5, 6, 7};
    int32       pointsToRead1 = 4;
    int32       pointsToRead2 = 3;
    int32       pointsRead;
    float64     timeout = 10.0;
    TaskHandle  taskHandle1 = 0;
    TaskHandle  taskHandle2 = 0;

    int         npins;
    std::string trash;
    timeval start, end,temp1,temp2;
    double tmp1, tmp2, tottime, AVGtime;
    int dataStored1[]={0, 0, 0, 0};
    int dataStored2[]={0, 0, 0};
    int tottrials;
    int kk, jj;
    int iold=0;
    int ii=1;
    int       POSX1,POSX2;

    ofstream myfile;
    ifstream myfile2;

//---------------------------------------------------
int main(int argc, char **argv)
```

```
{
    float64     min = -5.0;
    float64     max = 5.0;

//------------DAQ------------------------------------------

//Set Logfile
    printf ("\n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n" );
    cout << "Enter Test Subject Id #:  (es. 1234): ";
    cin >> FileName ;
    cout << "Enter Task File Name:  (es. A): ";
    cin >> TaskFile ;

    //ifstream myfile2;
    myfile2.open(TaskFile);
    if (myfile2.is open()) {printf("Task File Opened \n");
    } else {
        printf("ERROR: Could not open Task File \n");
        return 0;
    }

    getline (myfile2,line);
    istringstream buffer(line);
    buffer >> tottrials;


    //ofstream myfile;
    myfile.open(FileName);

    myfile <<"            Subject              "<< (FileName) <<" \n";
    myfile <<"            Task File          "<< TaskFile <<" \n"  ;
    myfile <<"            Total Number of Trials :  "<< tottrials <<" \n"  ;

    //Set NIDAQ cards
    printf ("Initializing... \n ");
    DAQmxErrChk (DAQmxBaseCreateTask("",&taskHandle1));
    DAQmxErrChk (DAQmxBaseCreateAIVoltageChan(taskHandle1,chan1,"",DAQmx_Val_Cfg_Default,min,max,DAQmx_Val_Volts,NULL));
    DAQmxErrChk (DAQmxBaseStartTask(taskHandle1));
    DAQmxErrChk (DAQmxBaseCreateTask("",&taskHandle2));
    DAQmxErrChk (DAQmxBaseCreateAIVoltageChan(taskHandle2,chan2,"",DAQmx_Val_Cfg_Default,min,max,DAQmx_Val_Volts,NULL));
    DAQmxErrChk (DAQmxBaseStartTask(taskHandle2));

//--------------------------------------------------------
    glutInit(&argc, argv);
    init();
    arVideoCapStart();
    glColor3f( 0.0, 0.0, 0.0 );
    argMainLoop( NULL, keyEvent, mainLoop);

Error:
              if( DAQmxFailed(error) )
                  DAQmxBaseGetExtendedErrorInfo(errBuff,2048);
              if( taskHandle1!=0 ) {
                  DAQmxBaseStopTask(taskHandle1);
                  DAQmxBaseClearTask(taskHandle1);
              }
              if( taskHandle2!=0 ) {
                  DAQmxBaseStopTask(taskHandle2);
                  DAQmxBaseClearTask(taskHandle2);
              }

          if( DAQmxFailed(error) )
          printf("DAQmxBase Error: %s\n",errBuff);

        myfile.close();
        myfile2.close();
}

void renderBitmapString(float x, float y,float z, void *font, char *String)
{
  char *c;
  glRasterPos3d(x,y,z);
  for (c=String; *c != '\0'; c++)
  {
    glutBitmapCharacter(font, *c);
  }
}

static void   keyEvent( unsigned char key, int x, int y)
{

    /* quit if the ESC key is pressed */
    if( key == 0x1b ) {
        cleanup();
        DAQmxBaseStopTask(taskHandle1);
        DAQmxBaseClearTask(taskHandle1);
        DAQmxBaseStopTask(taskHandle2);
        DAQmxBaseClearTask(taskHandle2);
        printf ("Test Completed ! \n");
         exit(0);
    }

    if( key == '+' ) {
     thresh++;
        printf("Threshold = %d): ", thresh);
        printf("\n");

    }
     if( key == '-' ) {
     thresh--;
```

```
        printf("Threshold = %d): ", thresh);
        printf("\n");

    }


}

/* main loop */
static void mainLoop(void)
{
    ARUint8         *dataPtr;
    ARMarkerInfo    *marker_info;
    int             marker_num;
    double          err;



    //------------------------DAQ MAIN LOOP

    if(ii>tottrials) {
        cleanup();
        DAQmxBaseStopTask(taskHandle1);
        DAQmxBaseClearTask(taskHandle1);
        DAQmxBaseStopTask(taskHandle2);
        DAQmxBaseClearTask(taskHandle2);
        printf ("Test Completed ! \n");
        myfile.close();
        myfile2.close();
        exit(0);
    }
    if(ii>iold){
        iold=ii;
        myfile <<"              Trial               "<< ii <<" \n"  ;
        myfile <<"% Trial "<< ii <<" \n"  ;

        if (ii==1) {
            printf ("\n Trial %u \n",ii);
            //cout << "Enter # of pins to move in this trial: ";
                    getline (myfile2,line);
                    istringstream buffer(line);
                    buffer >> npins;
        }else{
            cout << "\n ------------ RESET PINS ------------ \n";

            printf ("\n Trial %u \n",ii);
            //cout << "Enter # of pins to move in this trial: ";
        }

        getline (myfile2,line);
        istringstream buffer(line);
        buffer >> int_line;

        myfile <<"             Pin Movements in Trial :  " << npins << " \n"  ;
        kk=0;
        tottime=0;
        AVGtime=0;
        dataStored1[0]=0;
        dataStored1[1]=0;
        dataStored1[2]=0;
        dataStored1[3]=0;
        dataStored2[0]=0;
        dataStored2[1]=0;
        dataStored2[2]=0;

        printf("Press RETURN to START \n");
        trash="trash";
        if(ii==1){
            std::getline(std::cin, trash);
            std::getline(std::cin, trash);
            gettimeofday(&start, NULL);
            printf("\n START ! \n");
        }else{
            std::getline(std::cin, trash);
            gettimeofday(&start, NULL);
            printf("\a");
            printf("\n START ! \n");
        }

    }

        DAQmxErrChk (DAQmxBaseReadAnalogF64(taskHandle1,pointsToRead1,timeout,DAQmx_Val_GroupByChannel,data1,
samplesPerChan1,&pointsRead,NULL));

        DAQmxErrChk (DAQmxBaseReadAnalogF64(taskHandle2,pointsToRead2,timeout,DAQmx_Val_GroupByChannel,data2,
samplesPerChan2,&pointsRead,NULL));


        Error:
    /*

                if( DAQmxFailed(error) )
                    DAQmxBaseGetExtendedErrorInfo(errBuff,2048);
                if( taskHandle1!=0 ) {
                    DAQmxBaseStopTask(taskHandle1);
                    DAQmxBaseClearTask(taskHandle1);
                }
                if( taskHandle2!=0 ) {
                    DAQmxBaseStopTask(taskHandle2);
```

```
                DAQmxBaseClearTask(taskHandle2);
            }

        if( DAQmxFailed(error) )
        printf("DAQmxBase Error: %s\n",errBuff);
    */


        for (int i=0; i<=3; i++){
            if (((data1[i] > 0.3) || (data1[i]<-0.3) ) && (dataStored1[i]==0))
            {
                printf("\a");
                gettimeofday(&end, NULL);
                tmp1 = start.tv_sec + (start.tv_usec / 1000000.0);
                tmp2 = end.tv_sec   + (end.tv_usec   / 1000000.0);
                double time1 = tmp2 - tmp1;
                if (data1[i]>-1.5 && data1[i]<-1.3) jj=0;
                if (data1[i]>-0.7 && data1[i]<-0.5) jj=1;
                if (data1[i]>0.5 && data1[i]<0.7) jj=2;
                if (data1[i]>1.1 && data1[i]<1.3) jj=3;
                if (data1[i]>1.7 && data1[i]<1.9) jj=4;
                if (data1[i]>2.5 && data1[i]<2.7) jj=5;
                if (data1[i]>3.5 && data1[i]<3.7) jj=6;




                if ((int_line/10 == datap[jj]) && ((int_line-(int_line/10)*10)==7-i)) {
                    myfile << setprecision(4) << (time1) <<" " << (datap[jj]) <<"   " << (7-i) <<" Correct \n"  ;
                }else{
                    myfile << setprecision(4) << (time1) <<" " << (datap[jj]) <<"   " << (7-i) <<" Wrong \n"  ;
                }

                dataStored1[i]=1;
                kk++;
                tottime=time1+tottime;
                AVGtime=time1/npins+AVGtime;
                gettimeofday(&start, NULL);
                getline (myfile2,line);
                istringstream buffer(line);
                buffer >> int_line;
                printf ("Pin Insertion Acquired %u / %u \n",kk,npins);
            }
        }
        for (int i=0; i<=2; i++){
            if (((data2[i] > 0.3) || (data2[i]<-0.3) ) && (dataStored2[i]==0))
            {
                printf("\a");
                gettimeofday(&end, NULL);
                tmp1 = start.tv_sec + (start.tv_usec / 1000000.0);
                tmp2 = end.tv_sec   + (end.tv_usec   / 1000000.0);
                double time1 = tmp2 - tmp1;
                if (data2[i]>-1.5 && data2[i]<-1.3) jj=0;
                if (data2[i]>-0.7 && data2[i]<-0.5) jj=1;
                if (data2[i]>0.5 && data2[i]<0.7) jj=2;
                if (data2[i]>1.1 && data2[i]<1.3) jj=3;
                if (data2[i]>1.7 && data2[i]<1.9) jj=4;
                if (data2[i]>2.5 && data2[i]<2.7) jj=5;
                if (data2[i]>3.5 && data2[i]<3.7) jj=6;

                if ((int_line/10 == datap[jj]) && ((int_line-(int_line/10)*10)==3-i)) {
                    myfile << setprecision(4) << (time1) <<" " << (datap[jj]) <<"   " << (3-i) <<" Correct \n"  ;
                }else{
                    myfile << setprecision(4) << (time1) <<" " << (datap[jj]) <<"   " << (3-i) <<" Wrong \n"  ;
                }

                dataStored2[i]=1;
                kk++;
                tottime=time1+tottime;
                AVGtime=time1/npins+AVGtime;
                gettimeofday(&start, NULL);
                getline (myfile2,line);
                istringstream buffer(line);
                buffer >> int_line;
                printf ("Pin Insertion Acquired %u / %u \n",kk,npins);
            }
        }
        if (kk==npins && ii==iold) {
            myfile <<"            Total Trial Time :           "<<(tottime) <<"\n";
            myfile <<"            Average Insertion Time :         "<<(AVGtime) <<"\n";
            ii=ii+1;
            npins=int_line;
            printf("\a");
            printf("\a");
            printf("\a");



        }

                POSX1=int_line/10;
                POSX2=int_line-(int_line/10)*10;

/* grab a vide frame */
    if( (dataPtr = (ARUint8 *)arVideoGetImage()) == NULL ) {
        arUtilSleep(2);
        return;
    }
```

```
    /* detect the markers in the video frame */
    if( arDetectMarker(dataPtr, thresh, &marker_info, &marker_num) < 0 ) {
        cleanup();
        exit(0);
    }

    argDrawMode2D();
    //argDispImage( dataPtr, 0,0 );
    argDispImage( arImage, 0, 0);


    arVideoCapNext();

    if( (err=arMultiGetTransMat(marker_info, marker_num, config)) < 0 ) {
        argSwapBuffers();
        return;
    }
    //printf("err = %f\n", err);
    if(err > 100.0 ) {
        argSwapBuffers();
        return;
    }

    argDrawMode3D();
    argDraw3dCamera( 0, 0 );
    glClearDepth( 1.0 );
    glClear(GL_DEPTH_BUFFER_BIT);
    draw( config->trans, config->marker[0].trans, 0 );

    argSwapBuffers();


    //-----------------------------------



}

static void init( void )
{
    ARParam  wparam;

    /* open the video path */
    if( arVideoOpen( vconf ) < 0 ) exit(0);
    /* find the size of the window */
    if( arVideoInqSize(&xsize, &ysize) < 0 ) exit(0);
    printf("Image size (x,y) = (%d,%d)\n", xsize, ysize);

    /* set the initial camera parameters */
    if( arParamLoad(cparam_name, 1, &wparam) < 0 ) {
        printf("Camera parameter load error !!\n");
        exit(0);
    }
    arParamChangeSize( &wparam, xsize, ysize, &cparam );
    arInitCparam( &cparam );
    printf("*** Camera Parameter ***\n");
    arParamDisp( &cparam );

    if( (config = arMultiReadConfigFile(config_name)) == NULL ) {
        printf("config data load error !!\n");
        exit(0);
    }

    /* open the graphics window */
    argInit( &cparam, 1.0, 1, 2, 1, 0 );

}

/* cleanup function called when program exits */
static void cleanup(void)
{
ofstream myfile;
ifstream myfile2 (TaskFile);
    arVideoCapStop();
    arVideoClose();
    argCleanup();

}

static void draw( double trans1[3][4], double trans2[3][4], int mode )
{
    double    gl_para[16];
    GLfloat   mat_ambient1[]    = {1.0, 1.0, 1.0, 1.0};
    GLfloat   mat_flash1[]      = {1.0, 0.0, 0.0, 1.0};
    GLfloat   mat_flash_shiny1[]= {50.0};
    GLfloat   light_position[]  = {100.0,-200.0,200.0,0.0};
    GLfloat   ambi[]            = {0.1, 0.1, 0.1, 0.1};
    GLfloat   lightZeroColor[]  = {0.9, 0.9, 0.9, 0.1};

    argDrawMode3D();
    argDraw3dCamera( 0, 0 );
    glEnable(GL_DEPTH_TEST);
    glDepthFunc(GL_LEQUAL);

    /* load the camera transformation matrix */
    glMatrixMode(GL_MODELVIEW);
    argConvGlpara(trans1, gl_para);
    glLoadMatrixd( gl_para );
    argConvGlpara(trans2, gl_para);
```

```
    glMultMatrixd( gl_para );

        glEnable(GL_LIGHTING);
        glEnable(GL_LIGHT0);
        glLightfv(GL_LIGHT0, GL_POSITION, light_position);
        glLightfv(GL_LIGHT0, GL_AMBIENT, ambi);
        glLightfv(GL_LIGHT0, GL_DIFFUSE, lightZeroColor);
        glMaterialfv(GL_FRONT, GL_SPECULAR, mat_flash1);
        glMaterialfv(GL_FRONT, GL_SHININESS, mat_flash_shiny1);
        glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient1);

    glMatrixMode(GL_MODELVIEW);


    glLineWidth( 15.0 );
    glTranslatef( 0.0, 0.0, 25.0 );
       glBegin(GL_LINE_STRIP);
       glVertex3f(POSX1*50-50,-75.0,-20);
       glVertex3f(POSX2*50-50,75.0,-20.0);
    glEnd();

    glDisable( GL_LIGHTING );

    glDisable( GL_DEPTH_TEST );
}
```

B.3 Experiment Software: Card Generator.m

This application is meant to generate four files. It generates the reference Card file that includes the total number of trials in the deck, the numbers of pin insertions for each trial, and the specifics on which pin goes were. This files is read by the two test applications and serves as a reference for determining if the insertion was correct or wrong, and how to acquire the data. This function, also generates three latex files that will later need to be compiled, and that will deliver a printable deck of Cards, an HMD friendly version of the Deck, and a reference file that was used in the preliminary testing to assess which hole was which. The code was written in matlab as follows.

B.3.1 Code:

```
%% This Program Creates four files containing the definition of the tasks to
%% be executed in the experiment. The files are generated as follows:

% ---####.txt ------
% 45    # of Cards
% 4     # of pins in trial #1
% 25    # pin and hole #
% .... and so on.

% -----####-C.tex and ####-CBlack.tex ####-Reference.tex---------
% theese files are LaTex Code, and produce an identical file, for the only
% difference that the firs uses white background and black fonts, while the
% other uses black background and white fonts. (the first is to be printed,
% while the second is used to be displayed full screen in the visor for the
% second part of the test. The last file generates the legend for the
% indicators ordered from 1 to 7.

%% The first file (###.txt) will need to be loaded in the DAQaquisition
%% applet and will verify whether the pin insetrions have been executed
%% correctly. The second and third files (###-C.tex and ###-CBlack.tex), will need to be 
compiled in a
%% LaTex environment, and will generate a .pdf file that will have to be
%% printed in order to have the phisical cards(###-C.tex) or displayed fullscreen in the 
visor(###-CBlack.tex).

%% NOTE:
% Immages in order to have a properly formatted Document must be: 110x70
% pixel .eps files. and must be named 1p.eps throug 7p.eps for the pins, 1h.eps through
% 7h.eps for the holes and arrow.eps for the arrow.

%% Massimiliano Di Capua SSL 2008
clear
clc
DeckName=input('Insert the cards deck name \n','s');
NofCards=input('\nInsert the number of cards in the deck \n (will be rounded to the 
closest Integer multiple of 6) \n');
    intwarning off;
    NofCards=int16(NofCards/6)*6;
DeckNameEXT=[DeckName];
DeckNameEXTTEX=[DeckName,'-.tex'];
DeckNameEXTTEXBlack=[DeckName,'-Black.tex'];
ReferenceDocName=[DeckName,'-Reference.tex'];
Deck = fopen(DeckNameEXT,'w');
TexDeck = fopen(DeckNameEXTTEX,'w');
TexDeckBlack = fopen(DeckNameEXTTEXBlack,'w');
ReferenceDoc= fopen(ReferenceDocName,'w');

%% Write on Cards
fprintf(Deck,'%d \n',NofCards);
fprintf(TexDeck,'\\documentclass{article} \n\\usepackage{graphicx} \n\\begin{document} 
\n\\begin{center} \n');
fprintf(TexDeckBlack,'\\documentclass{article} \n\\usepackage{graphicx} \n\\usepackage 
```

```
{color} \n\\color{white} \n\\begin{document} \n\\pagecolor{black} \n\\begin{center}\n TEST ↙
\n \\newpage \n');
fprintf(ReferenceDoc,'\\documentclass{article} \n\\usepackage{graphicx} \n\\begin ↙
{document} \n\\begin{center} \n');
for ii=1:7
    fprintf(ReferenceDoc,'\\includegraphics{%dp.eps} \n',ii);
    fprintf(ReferenceDoc,'\\includegraphics{Arrow.eps} \n');
    fprintf(ReferenceDoc,'\\includegraphics{%dh.eps} \n',ii);
end
SubCards=[0 0 0 0 0 0 0];

for i=1:NofCards

    TaskNumb=randint(1,1,[2,7]);

    SubCardsID=0;
    while SubCardsID==0
        if SubCards(TaskNumb)<NofCards/6
            SubCards(TaskNumb)=SubCards(TaskNumb)+1;
            SubCardsID=1;
        else
            TaskNumb=randint(1,1,[2,7]);
        end
    end

    Pin=randperm(7);
    Hole=randperm(7);

    fprintf(Deck,'%d \n', TaskNumb);
    if i>1
        fprintf(TexDeck,'\n\\newpage \n\n');
        fprintf(TexDeckBlack,'\n\\newpage \n\n');
    end
    for ii=1:TaskNumb
        fprintf(Deck,'%d%d \n', Pin(ii), Hole(ii));

        fprintf(TexDeck,'\\includegraphics{%dp.eps} \n',Pin(ii));
        fprintf(TexDeck,'\\includegraphics{Arrow.eps} \n');
        fprintf(TexDeck,'\\includegraphics{%dh.eps} \n',Hole(ii));

        fprintf(TexDeckBlack,'\\includegraphics{%dbp.eps} \n',Pin(ii));
        fprintf(TexDeckBlack,'\\includegraphics{Arrowb.eps} \n');
        fprintf(TexDeckBlack,'\\includegraphics{%dbh.eps} \n',Hole(ii));
    end

end
fprintf(ReferenceDoc,'\\end{center} \n\\end{document}');
fprintf(TexDeck,'\\end{center} \n\\end{document}');
fprintf(TexDeckBlack,'\\end{center} \n\\end{document}');
fclose all
```

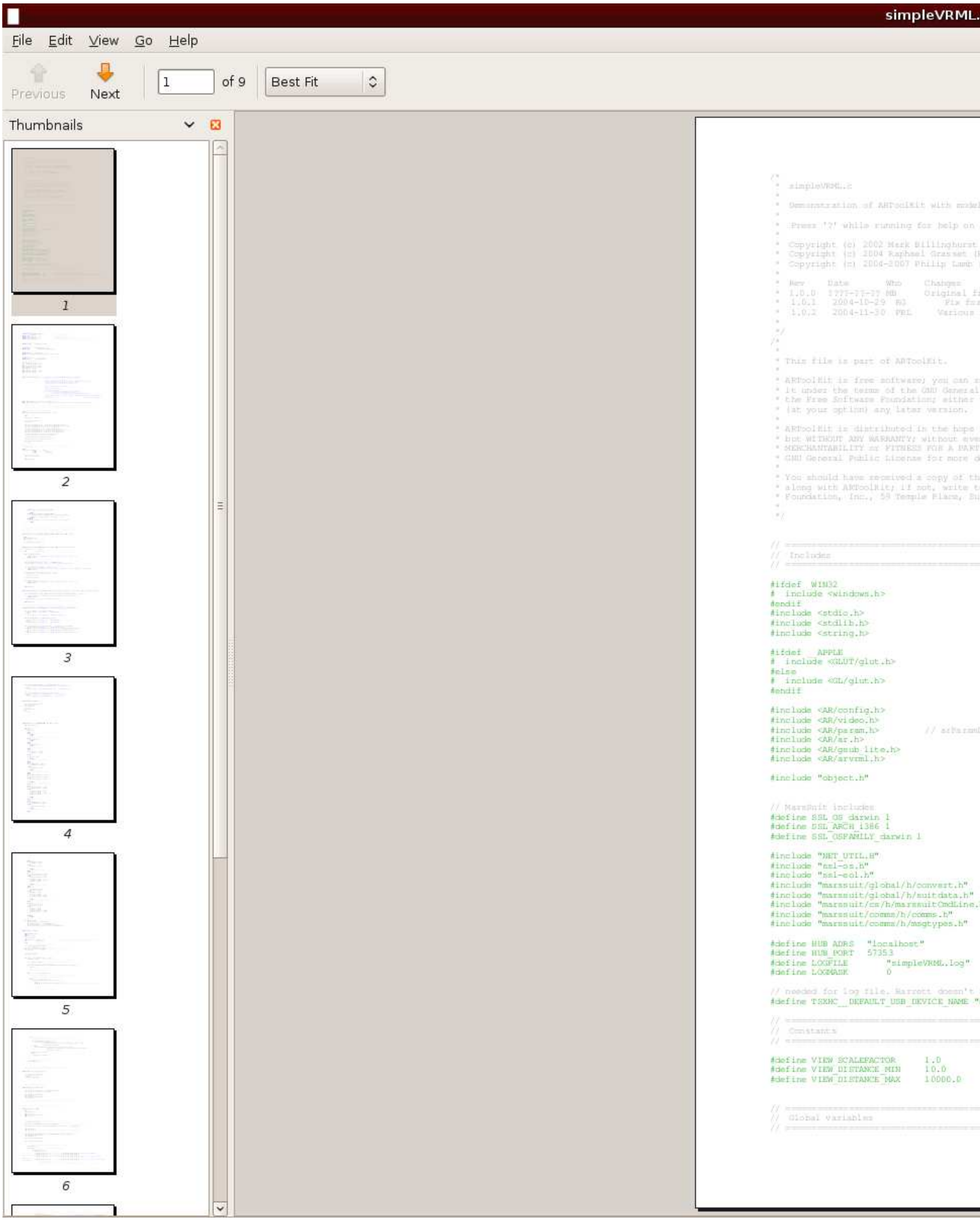
B.4 MX-2 Operations Software: MX-2-OPS.cpp

This application allows the user to access the suit status information through the RCL, and display:

- Text
- 2D overlays
- 3D GL overlays
- 3D VRML overlays

Although still in the early development phase, this application is able to recognize multiple markers at the same time, display object centered text and window centered text. It is also capable of displaying real time values read from the suit sensors, etc. This application will be further developed in the future to accommodate several other features as video playback capability and integration with the MX-2 speech recognition software. This C++ application uses many of the SSL libraries and its specific for MAC OS-X.

B.4.1 Code:



simpleVRML.
File
Edit
View
Go
Help
Previous
Next
1
of 9
Best Fit
Thumbnails
1
2
3
4
5
6
#ifdef WIN32
include <windows.h>
#endif
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#ifdef __APPLE__
include <GLUT/glut.h>
#else
include <GL/glut.h>
#endif
#include <AR/config.h>
#include <AR/video.h>
#include <AR/param.h>
#include <AR/ar.h>
#include <AR/gsub_lite.h>
#include <AR/arvrml.h>
#include "object.h"
#define SSL_OS_darwin 1
#define SSL_ARCH_i386 1
#define SSL_OSFAMILY_darwin 1
#include "NET_UTIL.H"
#include "ssl-os.h"
#include "ssl-eol.h"
#include "marssuit/global/h/convert.h"
#include "marssuit/global/h/suitdata.h"
#include "marssuit/cs/h/marssuitCmdLine.
#include "marssuit/comms/h/comms.h"
#include "marssuit/comms/h/msgtypes.h"
#define HUB_ADRS "localhost"
#define HUB_PORT 57353
#define LOGFILE "simpleVRML.log"
#define LOGMASK 0
#define TSXMC__DEFAULT_USB_DEVICE_NAME "
#define VIEW_SCALEFACTOR 1.0
#define VIEW_DISTANCE_MIN 10.0
#define VIEW_DISTANCE_MAX 10000.0

```
// Preferences.
static int prefWindowed = FALSE;
static int prefWidth = 800;                 // Fullscreen mode width.
static int prefHeight = 600;              // Fullscreen mode height.
static int prefDepth = 32;                  // Fullscreen mode bit depth.
static int prefRefresh = 0;                 // Fullscreen mode refresh rate. Set to 0 to use default rate.


// Image acquisition.
static ARUint8      *gARTImage = NULL;

// Marker detection.
static int          gARTThreshhold = 100;
static long             gCallCountMarkerDetect = 0;

// Drawing.
static ARParam      gARTCparam;
static ARGL_CONTEXT_SETTINGS_REF gArglSettings = NULL;

// Object Data.
static ObjectData_T     *gObjectData;
static int          gObjectDataCount;

// State variables
int CameraView = TRUE;
int TextDisplayMoving = TRUE;
int TextDisplayFixed = TRUE;
int CleanModeActive  = FALSE;
int ShowModel  = TRUE;
int ShowHelp  =FALSE;
int DisplaySuitStatus = FALSE;
int MoreInfo = FALSE;


// Text Variables
char *DisplaySteadyString[17] = { "WELCOME TO THE SSL's AUGMENTED REALITY SIMULATOR:",
                                  " ",
                                  "-----------------------------------------------",
                                  "This program simulates our vision for future space exploration.",
                                  "What you are about to see here is exactly what",
                                  "you would see if you were in our MX-2 Suit !",
                                  " ",
                                  "So... How about a little Game ?",
                                  " ",
                                  "This is how it works:",
                                  "1. flip two cards",
                                  "2. what do you see? Are the two objects identical?",
                                  "       - If so put the two cards away! Good Job !!",
                                  "       - If not, turn them back and try again with another set.",
                                  "3. The game is over when you've found all the pairs and there are no cards left" ,
                                  " ",
                                  "Have Fun !!!                                          (Press ''h'' for Help)" };


short int Xtextobj=0., Ytextobj=0., Ztextobj=-15.;
char *ObjectFixedString[6] = {"Subaru Impreza","Galileo","Hubble","ISS","Space Shuttle","Tie Fighter"};

// ============================================================================
//  RCL
// ============================================================================

/* incoming msg data is in NETWORK ORDER */
void
msgTelemetry(marssuitComms_SuitTelemetry_t *data)
{
    bool        b;

    assert(data != NULL);

    b = suitData_LockMutex();
    assert(b);

    suitData.sampleTime.tv_sec = ntohl(data->sampleTime_seconds);
    suitData.sampleTime.tv_nsec = ntohl(data->sampleTime_nanoseconds);
    suitData.pony1Pressure  = ntohf(data->pony1Pressure);
    suitData.pony2Pressure  = ntohf(data->pony2Pressure);
    suitData.pneumaSealPressure  = ntohf(data->pneumaSealPressure);
    suitData.ambientPressure  = ntohf(data->ambientPressure);
    suitData.backpackPressure  = ntohf(data->backpackPressure);
    suitData.suitPressure  = ntohf(data->suitPressure);
    suitData.heartRate  = ntohf(data->heartRate);

    suitData.digital_x  = ntohs(data->digital_x);
    suitData.digital_y  = ntohs(data->digital_y);

    b = suitData_UnlockMutex();
    assert(b);
}

/* deal with all incoming messages. NOTE that all incoming data is in
   NETWORK ORDER!!! */
void
msgReceiver(rclMsgID_t          msgID,
            void                *data,
            size_t              dataLength)
{
    assert(data != NULL);
    assert(0 < dataLength);

    switch (msgID)
```

```
    {
        case MARSSUIT_MESSAGEID_SUITCOMMANDS:
            /* should not get - ignore */
            break;
        case MARSSUIT_MESSAGEID_SUITTELEMETRY:
            assert(dataLength == sizeof(marssuitComms_SuitTelemetry_t));
            msgTelemetry((marssuitComms_SuitTelemetry_t *)data);
            break;
        default:
            fprintf(stderr, "Got invalid msg: msgid = %d" EOL, msgID);
            break;
    }
}


// ============================================================================
//  Functions
// ============================================================================

// This Function Displays a String n an OpenGL window
void renderBitmapString(float x, float y,float z, void *font, char *String)
{
  char *c;
  glRasterPos3d(x,y,z);
  for (c=String; *c != '\0'; c++)
  {
   glutBitmapCharacter(font, *c);
  }
}

static int setupCamera(const char *cparam_name, char *vconf, ARParam *cparam)
{
    ARParam          wparam;
    int              xsize, ysize;

    // Open the video path.
    if (arVideoOpen(vconf) < 0) {
        fprintf(stderr, "setupCamera(): Unable to open connection to camera.\n");
        return (FALSE);
    }

    // Find the size of the window.
    if (arVideoInqSize(&xsize, &ysize) < 0) return (FALSE);
    fprintf(stdout, "Camera image size (x,y) = (%d,%d)\n", xsize, ysize);

    // Load the camera parameters, resize for the window and init.
    if (arParamLoad(cparam_name, 1, &wparam) < 0) {
        fprintf(stderr, "setupCamera(): Error loading parameter file %s for camera.\n", cparam_name);
         return (FALSE);
    }
    arParamChangeSize(&wparam, xsize, ysize, cparam);
    fprintf(stdout, "*** Camera Parameter ***\n");
    arParamDisp(cparam);

    arInitCparam(cparam);

    if (arVideoCapStart() != 0) {
        fprintf(stderr, "setupCamera(): Unable to begin camera data capture.\n");
        return (FALSE);
    }

    return (TRUE);
}

static int setupMarkersObjects(char *objectDataFilename, ObjectData_T **objectDataRef, int *objectDataCountRef)
{
    // Load in the object data - trained markers and associated bitmap files.
    if ((*objectDataRef = read_VRMLdata(objectDataFilename, objectDataCountRef)) == NULL) {
         fprintf(stderr, "setupMarkersObjects(): read_VRMLdata returned error !!\n");
         return (FALSE);
    }
    printf("Object count = %d\n", *objectDataCountRef);

    return (TRUE);
}

// Report state of ARToolKit global variables arFittingMode,
// arImageProcMode, arglDrawMode, arTemplateMatchingMode, arMatchingPCAMode.
static void debugReportMode(const ARGL_CONTEXT_SETTINGS_REF arglContextSettings)
{
    if (arFittingMode == AR_FITTING_TO_INPUT) {
        fprintf(stderr, "FittingMode (Z): INPUT IMAGE\n");
    } else {
        fprintf(stderr, "FittingMode (Z): COMPENSATED IMAGE\n");
    }

    if (arImageProcMode == AR_IMAGE_PROC_IN_FULL) {
        fprintf(stderr, "ProcMode (X)   : FULL IMAGE\n");
    } else {
        fprintf(stderr, "ProcMode (X)   : HALF IMAGE\n");
    }

    if (arglDrawModeGet(arglContextSettings) == AR_DRAW_BY_GL_DRAW_PIXELS) {
        fprintf(stderr, "DrawMode (C)   : GL_DRAW_PIXELS\n");
    } else if (arglTexmapModeGet(arglContextSettings) == AR_DRAW_TEXTURE_FULL_IMAGE) {
        fprintf(stderr, "DrawMode (C)   : TEXTURE MAPPING (FULL RESOLUTION)\n");
    } else {
        fprintf(stderr, "DrawMode (C)   : TEXTURE MAPPING (HALF RESOLUTION)\n");
    }
```

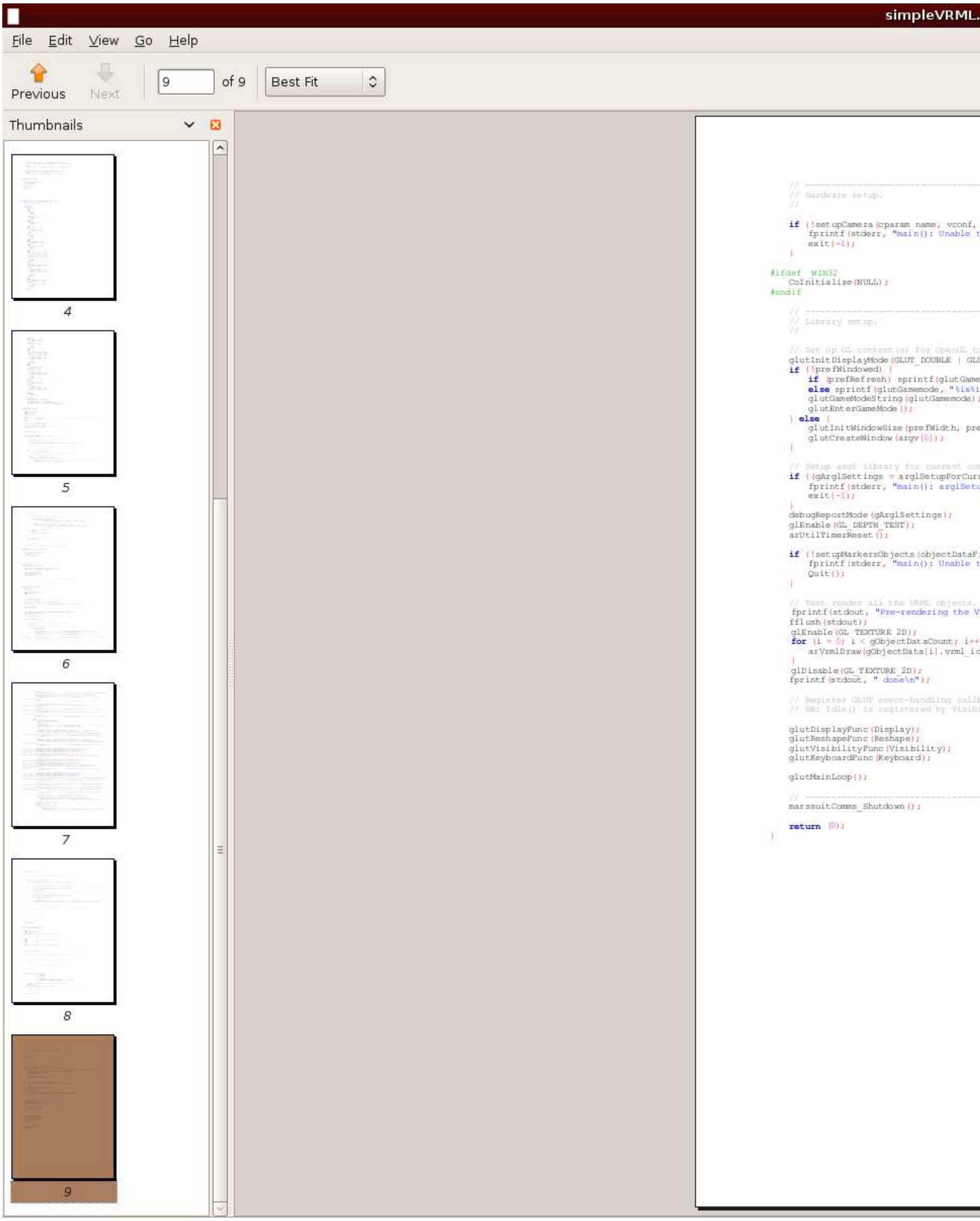
```
    if (arTemplateMatchingMode == AR_TEMPLATE_MATCHING_COLOR) {
        fprintf(stderr, "TemplateMatchingMode (M)    : Color Template\n");
    } else {
        fprintf(stderr, "TemplateMatchingMode (M)    : BW Template\n");
    }

    if (arMatchingPCAMode == AR_MATCHING_WITHOUT_PCA) {
        fprintf(stderr, "MatchingPCAMode (P)    : Without PCA\n");
    } else {
        fprintf(stderr, "MatchingPCAMode (P)    : With PCA\n");
    }
}

static void Quit(void)
{
    arglCleanup(gArglSettings);
    arVideoCapStop();
    arVideoClose();
#ifdef _WIN32
    CoUninitialize();
#endif
    exit(0);
}




static void Keyboard(unsigned char key, int x, int y)
{
    int threshChange = 0;

    switch (key) {
    //====== QUIT
        case 0x1B:
        case 'Q':
        case 'q':
            Quit();
            break;
    //====== Increase Decrease Contrast Threshold
        case '-':
            threshChange = -5;
            break;
        case '+':
        case '=':
            threshChange = +5;
            break;
    //===== Show/Hide camera view
        case 'c':
        case 'C':
        if (CameraView == TRUE){
            CameraView = FALSE;

        }else{
        CameraView = TRUE;
        }
            break;
        //===== Windowed/Full-Screen
        case 'w':
        case 'W':
        if (prefWindowed == TRUE){
            prefWindowed = FALSE;
        }else{
            prefWindowed = TRUE;
        }
            break;
    //===== Show/Hide Model Solidal Text
        case 't':
        case 'T':
        if (TextDisplayMoving == TRUE){
            TextDisplayMoving = FALSE;
        }else{
            TextDisplayMoving = TRUE;
        }
            break;
    //===== Show/Hide Screen Solidal Text
        case 's':
        case 'S':
        if (TextDisplayFixed == TRUE){
            TextDisplayFixed = FALSE;
        }else{
            TextDisplayFixed = TRUE;
            MoreInfo = FALSE;
            DisplaySuitStatus = FALSE;
            ShowHelp = FALSE;
        }
            break;
    //===== Clean Mode On/Off
        case 'n':
        case 'N':
        if (CleanModeActive== TRUE){
            CleanModeActive = FALSE;
        }else{
            CleanModeActive = TRUE;
        }
            break;
    //===== Show/Hide Model
        case 'm':
```

```
        case 'M':
        if (ShowModel== TRUE){
            ShowModel = FALSE;
        }else{
            ShowModel = TRUE;
        }
            break;
    //===== Show/Hide Suit Status
        case 'a':
        case 'A':
        if (DisplaySuitStatus== TRUE){
            DisplaySuitStatus = FALSE;
        }else{
            DisplaySuitStatus = TRUE;
            MoreInfo = FALSE;
            TextDisplayFixed = FALSE;
            ShowHelp = FALSE;
        }
            break;
    //===== Show/Hide Display Help
        case 'h':
        case 'H':
            if (ShowHelp== TRUE){
            ShowHelp = FALSE;
        }else{
            ShowHelp = TRUE;
            MoreInfo = FALSE;
            DisplaySuitStatus = FALSE;
            TextDisplayFixed = FALSE;
        }
            break;
            //===== Show/Hide More Info on Object
        case 'I':
        case 'i':
            if (MoreInfo== TRUE){
            MoreInfo = FALSE;
        }else{
            MoreInfo = TRUE;
            ShowHelp = FALSE;
            DisplaySuitStatus = FALSE;
            TextDisplayFixed = FALSE;
        }

        default:
            break;
        }

    if (threshChange) {
        gARTThreshhold += threshChange;
        if (gARTThreshhold < 0) gARTThreshhold = 0;
        if (gARTThreshhold > 255) gARTThreshhold = 255;
        printf("Threshhold changed to %d.\n", gARTThreshhold);
    }
}

static void Idle(void)
{
    static int ms_prev;
    int ms;
    float s_elapsed;
    ARUint8 *image;

    ARMarkerInfo    *marker_info;                   // Pointer to array holding the details of detected markers.
     int             marker_num;                        // Count of number of markers detected.
     int             i, j, k;

    // Find out how long since Idle() last ran.
    ms = glutGet(GLUT_ELAPSED_TIME);
    s_elapsed = (float)(ms - ms_prev) * 0.001;
    if (s_elapsed < 0.01f) return; // Don't update more often than 100 Hz.
    ms_prev = ms;

    // Update drawing.
    arVrmlTimerUpdate();

    // Grab a video frame.
    if ((image = arVideoGetImage()) != NULL) {
        gARTImage = image; // Save the fetched image.

        gCallCountMarkerDetect++; // Increment ARToolKit FPS counter.

        // Detect the markers in the video frame.
        if (arDetectMarker(gARTImage, gARTThreshhold, &marker_info, &marker_num) < 0) {
            exit(-1);
        }

        // Check for object visibility.

        for (i = 0; i < gObjectDataCount; i++) {

            // Check through the marker_info array for highest confidence
            // visible marker matching our object's pattern.
            k = -1;
            for (j = 0; j < marker_num; j++) {
                if (marker_info[j].id == gObjectData[i].id) {
                    if( k == -1 ) k = j; // First marker detected.
                    else if (marker_info[k].cf < marker_info[j].cf) k = j; // Higher confidence marker detected.
                }
            }
```

```
            if (k != -1) {
                // Get the transformation between the marker and the real camera.
                //fprintf(stderr, "Saw object %d.\n", i);
                if (gObjectData[i].visible == 0) {
                    arGetTransMat(&marker_info[k],
                                  gObjectData[i].marker_center, gObjectData[i].marker_width,
                                  gObjectData[i].trans);
                } else {
                    arGetTransMatCont(&marker_info[k], gObjectData[i].trans,
                                      gObjectData[i].marker_center, gObjectData[i].marker_width,
                                      gObjectData[i].trans);
                }
                gObjectData[i].visible = 1;
            } else {
                gObjectData[i].visible = 0;
            }
        }

        // Tell GLUT the display has changed.
        glutPostRedisplay();
    }
}

//
//  This function is called on events when the visibility of the
//  GLUT window changes (including when it first becomes visible).
//
static void Visibility(int visible)
{
    if (visible == GLUT_VISIBLE) {
        glutIdleFunc(Idle);
    } else {
        glutIdleFunc(NULL);
    }
}

//
//  This function is called when the
//  GLUT window is resized.
//
static void Reshape(int w, int h)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glViewport(0, 0, (GLsizei) w, (GLsizei) h);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    // Call through to anyone else who needs to know about window sizing here.
}

//
// This function is called when the window needs redrawing.
//
static void Display(void)
{
    int i, nol;
    GLdouble p[16];
    GLdouble m[16];
    char StatusString2display[50];
    bool mutex_ok;


    // Select correct buffer for this context.
    glDrawBuffer(GL_BACK);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // Clear the buffers for new frame.

    if (CameraView == TRUE) arglDispImage(gARTImage, &gARTCparam, 1.0, gArglSettings);    // zoom = 1.0.

    arVideoCapNext();
    gARTImage = NULL; // Image data is no longer valid after calling arVideoCapNext().

    // Projection transformation.
    arglCameraFrustumRH(&gARTCparam, VIEW_DISTANCE_MIN, VIEW_DISTANCE_MAX, p);
    glMatrixMode(GL_PROJECTION);
    glLoadMatrixd(p);
    glMatrixMode(GL_MODELVIEW);


    if (ClearModeActive==FALSE){


        // This section displays green text, solidal to the window
        glLoadIdentity();
        gluLookAt(0.0, 0.0, 250,0.0,0.0,0.0,0.0f,1.0f,0.0f);

                if(ShowHelp==TRUE){
                    glColor3f(1,1,1);
                    renderBitmapString(-110.,70.-10.*6,0 , GLUT_BITMAP_TIMES_ROMAN_24, "Keybord Input Help");
                    renderBitmapString(-110.,70.-10.*7,0 , GLUT_BITMAP_TIMES_ROMAN_24,
"-----------------------------------------------------" );
                    renderBitmapString(-110.,70.-10.*8,0 , GLUT_BITMAP_TIMES_ROMAN_24, "Esc / Q (q) =========>  Quit
Program");
                    renderBitmapString(-110.,70.-10.*9,0 , GLUT_BITMAP_TIMES_ROMAN_24,  "A (a)    ============>  Show/Hide
Suit Status");
                    /*if (MoreInfo==FALSE){glColor3f(1,0,0);}
                        renderBitmapString(-110.,70.-10.*8,0 , GLUT_BITMAP_TIMES_ROMAN_24,  "I (i)    ============>
```

```
Show/Hide More Object Info");*/
                glColor3f(1,1,1);
                sprintf(StatusString2display,"+ / -  =============>  Increase/Decrease Contrast (%i)", gARTThreshhold);
                renderBitmapString(-110.,70.-10.*10,0 , GLUT_BITMAP_TIMES_ROMAN_24, StatusString2display);
                if (CameraView==FALSE){glColor3f(1,0,0);}
                    renderBitmapString(-110.,70.-10.*11,0 , GLUT_BITMAP_TIMES_ROMAN_24, "C (c)  =============>
Hide/Show Webcam View");
                glColor3f(1,1,1);
                //renderBitmapString(-110.,70.-10.*11,0 , GLUT_BITMAP_TIMES_ROMAN_24, "W (w)  =============>
Full-Screen/Window");
                if (TextDisplayMoving==FALSE){glColor3f(1,0,0);}
                    renderBitmapString(-110.,70.-10.*12,0 , GLUT_BITMAP_TIMES_ROMAN_24, "T (t)  =============>
Show/Hide Model fixed Text");
                glColor3f(1,1,1);
                renderBitmapString(-110.,70.-10.*13,0 , GLUT_BITMAP_TIMES_ROMAN_24, "S (s)  =============>  Show/Hide
Window Fixed Text");
                if (ShowModel==FALSE){glColor3f(1,0,0);}
                    renderBitmapString(-110.,70.-10.*14,0 , GLUT_BITMAP_TIMES_ROMAN_24, "M (m)  =============>
Show/Hide Model");
                glColor3f(1,1,1);
                renderBitmapString(-110.,70.-10.*15,0 , GLUT_BITMAP_TIMES_ROMAN_24, "H (h)  =============>  This Help!"
);
            }else{
                if (DisplaySuitStatus == TRUE){
                    mutex_ok = suitData_LockMutex();
                    assert(mutex_ok);
                    glColor3f(1,1,1);
                    renderBitmapString(-110.,70.-10.*7,0 , GLUT_BITMAP_TIMES_ROMAN_24, "        Suit Status Monitor");
                    renderBitmapString(-110.,70.-10.*8,0 , GLUT_BITMAP_TIMES_ROMAN_24,
"------------------------------------");

                    if (MARSSUIT_CONVERT__PONY1PRESSURE(suitData.pony1Pressure)<=1000 ){glColor3f(1,0.65,0);}
                    if (MARSSUIT_CONVERT__PONY1PRESSURE(suitData.pony1Pressure)<=500 ){glColor3f(1,0,0);}
                        sprintf(StatusString2display,"Pony 1 Pressure %4.2f  psi", MARSSUIT_CONVERT__PONY1PRESSURE(
suitData.pony1Pressure));
                        renderBitmapString(-110.,70.-10.*9,0 , GLUT_BITMAP_TIMES_ROMAN_24,StatusString2display);
                    glColor3f(1,1,1);

                    if (MARSSUIT_CONVERT__PONY2PRESSURE(suitData.pony2Pressure)<=1000){glColor3f(1,0.65,0);}
                    if (MARSSUIT_CONVERT__PONY2PRESSURE(suitData.pony2Pressure)<=500){glColor3f(1,0,0);}
                        sprintf(StatusString2display,"Pony 2 Pressure %4.2f  psi",  MARSSUIT_CONVERT__PONY2PRESSURE(
suitData.pony2Pressure));
                        renderBitmapString(-110.,70.-10.*10,0 , GLUT_BITMAP_TIMES_ROMAN_24, StatusString2display);
                    glColor3f(1,1,1);

                    if (MARSSUIT_CONVERT__PNEUMASEALPRESSURE(suitData.pneumaSealPressure)<=45 &&
MARSSUIT_CONVERT__PNEUMASEALPRESSURE(suitData.pneumaSealPressure)>=60){glColor3f(1,0.65,0);}
                    if (MARSSUIT_CONVERT__PNEUMASEALPRESSURE(suitData.pneumaSealPressure)<=40 &&
MARSSUIT_CONVERT__PNEUMASEALPRESSURE(suitData.pneumaSealPressure)>=55){glColor3f(1,0,0);}
                        sprintf(StatusString2display,"Pneuma Seal Pressure %4.2f  psi",
MARSSUIT_CONVERT__PNEUMASEALPRESSURE(suitData.pneumaSealPressure));
                        renderBitmapString(-110.,70.-10.*11,0 , GLUT_BITMAP_TIMES_ROMAN_24, StatusString2display);
                    glColor3f(1,1,1);

                    if (MARSSUIT_CONVERT__AMBIENTPRESSURE(suitData.ambientPressure)<=14 ||
MARSSUIT_CONVERT__AMBIENTPRESSURE(suitData.ambientPressure)>=40){glColor3f(1,0.65,0);}
                    if (MARSSUIT_CONVERT__AMBIENTPRESSURE(suitData.ambientPressure)<=10 ||
MARSSUIT_CONVERT__AMBIENTPRESSURE(suitData.ambientPressure)>=50){glColor3f(1,0,0);}
                        sprintf(StatusString2display,"Ambient Pressure %4.2f  psi (Depth: %4.1f feet)",
MARSSUIT_CONVERT__AMBIENTPRESSURE(suitData.ambientPressure),  (MARSSUIT_CONVERT__AMBIENTPRESSURE(suitData.
ambientPressure)-14.7)/0.432);
                        renderBitmapString(-110.,70.-10.*12,0 , GLUT_BITMAP_TIMES_ROMAN_24, StatusString2display);
                    glColor3f(1,1,1);

                    if (MARSSUIT_CONVERT__BACKPACKPRESSURE(suitData.backpackPressure)<=1 ||
MARSSUIT_CONVERT__BACKPACKPRESSURE(suitData.backpackPressure)>=4){glColor3f(1,0.65,0);}
                    if (MARSSUIT_CONVERT__BACKPACKPRESSURE(suitData.backpackPressure)<=0.5 ||
MARSSUIT_CONVERT__BACKPACKPRESSURE(suitData.backpackPressure)>=4.5){glColor3f(1,0,0);}
                        sprintf(StatusString2display,"Backpack Pressure %4.2f  psi", MARSSUIT_CONVERT__BACKPACKPRESSURE
(suitData.backpackPressure));
                        renderBitmapString(-110.,70.-10.*13,0 , GLUT_BITMAP_TIMES_ROMAN_24, StatusString2display);
                    glColor3f(1,1,1);

                    if (MARSSUIT_CONVERT__SUITPRESSURE(suitData.suitPressure)<=2 || MARSSUIT_CONVERT__SUITPRESSURE(
suitData.suitPressure)>=3){glColor3f(1,0.65,0);}
                    if (MARSSUIT_CONVERT__SUITPRESSURE(suitData.suitPressure)<=1.5 || MARSSUIT_CONVERT__SUITPRESSURE(
suitData.suitPressure)>=3.5){glColor3f(1,0,0);}
                        sprintf(StatusString2display,"Suit Pressure %4.2f  psi",  MARSSUIT_CONVERT__SUITPRESSURE(
suitData.suitPressure));
                        renderBitmapString(-110.,70.-10.*14,0 , GLUT_BITMAP_TIMES_ROMAN_24, StatusString2display);
                    glColor3f(1,1,1);

                    if (suitData.heartRate>=100 ||  suitData.heartRate<=60){glColor3f(1,0.65,0);}
                    if (suitData.heartRate>=120 ||  suitData.heartRate<=50){glColor3f(1,0,0);}
                        sprintf(StatusString2display,"Heart Rate %3.0f  bps",  suitData.heartRate);
                        renderBitmapString(-110.,70.-10.*15,0 , GLUT_BITMAP_TIMES_ROMAN_24, StatusString2display);

                    mutex_ok = suitData_UnlockMutex();
                    assert(mutex_ok);
                }else{
                    if (TextDisplayFixed == TRUE){
                        for (nol=0; nol < 16; nol++){
                            glColor3f(1,1,1);
                            renderBitmapString(-110.,70.-10.*nol,0 , GLUT_BITMAP_TIMES_ROMAN_24, DisplaySteadyString[nol
]);
                        }
                    }
                }
            }
```

```
    // Viewing transformation.
    glLoadIdentity();

    // Lighting and geometry that moves with the camera should go here.
    // (I.e. must be specified before viewing transformations.)
    //none

        for (i = 0; i < gObjectDataCount; i++) {

            if ((gObjectData[i].visible != 0) && (gObjectData[i].vrml_id >= 0)) {

                // Calculate the camera position for the object and draw it.
                // Replace VIEW_SCALEFACTOR with 1.0 to make one drawing unit equal to 1.0 ARToolKit units (usually
millimeters).
                arglCameraViewRH(gObjectData[i].trans, m, VIEW_SCALEFACTOR);
                glLoadMatrixd(m);

                // All lighting and geometry to be drawn relative to the marker goes here.
                //fprintf(stderr, "About to draw object %i\n", i);
                if(ShowModel == TRUE) arVrmlDraw(gObjectData[i].vrml_id);

                if (TextDisplayMoving==TRUE){
                    // Display text in green
                    glColor3f(0,1,0);
                    renderBitmapString(Xtextobj,Ytextobj,Ztextobj , GLUT_BITMAP_TIMES_ROMAN_24, ObjectFixedString[i]);
                /*if(MoreInfo == TRUE){
                            TextDisplayFixed = FALSE;
                            for (nol=0; nol < 16; nol++){
                                glColor3f(1,1,1);
                                renderBitmapString(-110.,70.-10.*nol,0 , GLUT_BITMAP_TIMES_ROMAN_24,
DisplaySteadyString[i+1][nol]);
                            }
                    }*/
                }
            }
        }

    }
    // Any 2D overlays go here.
    //none

    glutSwapBuffers();

}

int main(int argc, char** argv)
{
    int i;
    char glutGamemode[32];
    const char *cparam_name = "Data/camera_para.dat";
    //
    // Camera configuration.
    //
#ifdef _WIN32
    char            *vconf = "Data\\WDM_camera_flipV.xml";
#else
    char            *vconf = "";
#endif
    char objectDataFilename[] = "Data/object_data_vrml";

    // ---------------------------------------------------------------------------
    // RCL & Suit Data inits

    suitData_Initialize();

    //ssl::marssuit::MarssuitCommandLineArguments *args =
    //  new ssl::marssuit::MarssuitCommandLineArguments(TEXHC__DEFAULT_USB_DEVICE_NAME);
    //FILE*                   logfile = NULL;

    /*int rc = args->GetProgramOptions(argc, argv, __DATE__, __TIME__);
    if (0 != rc) {
        // have an actual error, or did user just ask for help?
        if (1 != rc)
        {
            printf(EOL "ERROR: main: unable parse command line." EOL);
        }
        return -1;
    }
    */

    /* init system, RCL and screens */
    if (marssuitComms_Init(LOGFILE,
                           LOGMASK,
                            HUB_ADRS,
                           HUB_PORT,
                           RCLCHANNELID__MARSSUIT_COMMANDS,
                           RCLCHANNELID__MARSSUIT_TELEMETRY) != GEN_ERR_NONE)
    {
        printf(EOL "ERROR: main: unable init communications" EOL);
        return -1;
    }
    marssuitComms_RegisterReceiver(msgReceiver);

    // ---------------------------------------------------------------------------
    // Library inits.
    //
    glutInit(&argc, argv);
```


simpleVRML.
File
Edit
View
Go
Help
Previous
Next
9
of 9
Best Fit
Thumbnails
4
5
6
7
8
9
// Hardware setup.
exit(-1);
#ifdef WIN32
CoInitialize(NULL);
#endif
// Library setup.
glutGameModeString(glutGamemode);
glutEnterGameMode();
} else {
glutCreateWindow(argv[0]);
exit(-1);
debugReportMode(gArglSettings);
glEnable(GL_DEPTH_TEST);
arUtilTimerReset();
Quit();
fflush(stdout);
glEnable(GL_TEXTURE_2D);
glDisable(GL_TEXTURE_2D);
fprintf(stdout, " done\n");
glutDisplayFunc(Display);
glutReshapeFunc(Reshape);
glutVisibilityFunc(Visibility);
glutKeyboardFunc(Keyboard);
glutMainLoop();
marssuitComms_Shutdown();
return (0);
}

Appendix C

Approved IRB

UNIVERSITY OF MARYLAND

INSTITUTIONAL REVIEW BOARD

2100 Lee Building
College Park, Maryland 20742-5121
301.405.4212 TEL 301.314.1475 FAX
irb@deans.umd.edu
www.umresearch.umd.edu/IRB

July 2, 2008

MEMORANDUM

Application Approval Notification

To:	Dr. David Akin Massimilliano Di Capua Department of Aerospace Engineering
From:	Roslyn Edson, M.S., CIP IRB Manager University of Maryland, College Park
Re:	**IRB Application Number: # 08-0313** **Project Title: "Augmented Reality for Space Application. Performance and Accuracy Impact on Simple Unrehearsed Tasks. Comparison between In and Out-of-Field-of-View Information Feed Methods"**
Approval Date:	**June 19, 2008**
Expiration Date:	**June 19, 2009**
Type of Application:	Initial
Type of Research:	Non-Exempt
Type of Review For Application:	Full Board Review **Degree of Risk**: No Greater than Minimal Risk

The University of Maryland, College Park Institutional Review Board (IRB) approved your IRB application. The research was approved in accordance with 45 CFR 46, the Federal Policy for the Protection of Human Subjects, and the University's IRB policies and procedures. Please reference the above-cited IRB application number in any future communications with our office regarding this research.

Recruitment/Consent: For research requiring written informed consent, the IRB-approved and stamped informed consent document is enclosed. The IRB approval expiration date has been stamped on the informed consent document. Please keep copies of the consent forms used for this research for three years after the completion of the research.

Continuing Review: If you intend to continue to collect data from human subjects or to analyze private, identifiable data collected from human subjects, after the expiration date for this approval (indicated above), you must submit a renewal application to the IRB Office at least 30 days before the approval expiration date.

INITIAL APPLICATION

UNIVERSITY OF MARYLAND, COLLEGE PARK
Institutional Review Board
Initial Application for Research Involving Human Subjects
Please complete this cover page AND provide all information requested in the attached instructions.

Name of Principal Investigator (PI) or Project Faculty Advisor David L. Akin **Tel. No** 301-405-1138

Name of Co-Investigator (Co-PI) - **Tel. No** -

Department or Unit Administering the Project Aerospace Engineering

E-Mail Address dakin@ssl.umd.edu **E-Mail Address of Co-** -

Where should the IRB send the approval letter? University of Maryland Building 382 College Park, MD 20742-2911

Name of Student Investigator Massimiliano Di Capua **Tel.** 202-489-2626

E-Mail Address of Student Investigator maxdicapua@yahoo.com

Check here if this is a student master's thesis ☒ **or a dissertation research project** ☐

Project Duration (mo/yr – mo/yr) 06-08 – 09/08

Project Title Augmented Reality for Space Application: Performance and Accuracy Impact on Simple Unrehearsed Tasks. Comparison Between In and Out-Of-Field-Of-View Information Feed Methods.

Sponsored Project Data	**Funding Agency**	-	**ORAA Proposal**	-

Vulnerable Populations: The proposed research will involve the following (Check all that apply): pregnant women ☐, human fetuses ☐, neonates ☐, minors/children ☐, prisoners ☐, students ☒, individuals with mental disabilities ☐, individuals with physical disabilities ☐

Exempt or Nonexempt (Optional): You may recommend your research for exemption or non-exemption by completing the appropriate box below. For exempt recommendation, list the numbers for the exempt category(s)

☐ **Exempt—List Exemption Category** *Or* ☒ **Non-Exempt**

If exempt, briefly describe the reason(s) for exemption. Your notation is a suggestion to the IRB Manager and IRB Co-Chairs.
-

5/22/08

Date David L. Akin Faculty Advisor

5/22/08

Date Massimiliano Di Capua Student Investigator

5/22/08

Darryll Pines, Department Chair

Date

REC'D MAY 2 2 2008

IRB Application

0. Title:

Augmented Reality for Space Application: Performance and Accuracy Impact on Simple Unrehearsed Tasks. Comparison Between In and Out-Of-Field-Of-View Information Feed Methods.

1. Abstract:

The purpose of this study is to quantify performance and accuracy benefits arising from the use of Augmented Reality Head Mounted Displays (HMD). In this experiment we will simulate a simple unrehearsed task by asking the subject to remove a specific peg and insert it in a different hole on a digital pin-hole board. The subject will not have any prior knowledge of the sequences to be executed and will receive all necessary information just before commencing the task. The experiment will be divided in two parts based on the information display method in use. In the first part, the subject will receive a deck of cards placed in close proximity to the digital pin-hole board, with all the instructions for the specific tasks written on the cards, while in the second, the subject will make use of a head mounted display that will overlay, a schematic of the task, on the pin-hole board. In both cases, data on task completion time and accuracy will be collected. Qualitative data will also be collected subsequent the experiment through a brief experiment response questionnaire.

The experiment will be performed at the Space Systems Lab in the Aerospace Engineering Department. Subject involvement will be completely voluntary, informed consent will be obtained before commencement of the experiment, and all subject data will be kept confidential.

2. Subject Selection:

Subjects will be adult (over 18 years of age) students, faculty or staff from the College of Engineering. The number of subjects in the study will be between 10 and 15. The subjects will not be selected for race, ethnic origin, religion, or any social or economic qualifications. Ineligibility to the study will be caused exclusively from physical impairment that is expected to reduce speed and accuracy in movements. Subjects will be recruited formally through email. The recruitment email will be sent exclusively to the SSL Faculty, Students and Staff. A copy of the recruitment email is included with this application. Study participation will be completely voluntary. Subjects can withdraw from the study at any time for any reason.

3. Procedures:

The experiment will be conducted in a single session of approximately 1.5 hours for each subject. Once the subject has been introduced to the study and all the necessary forms have been signed, the experiment will begin. The subject will be asked to seat in a workstation where the equipment has been placed. The test subject will be then invited to find a comfortable position and will be encouraged to reposition the equipment as desired. Equipment can be repositioned during the experiment at any time. Changes in equipment configuration will be recorded by the Principal Investigator by taking a picture of the setup. The subject will be asked to step out the field of view of the camera in order to protect their privacy.

Experiment Setup:

The study is divided in two parts:

In part A, the subject will receive information on the task from a deck of cards placed in proximity to the digital pin-hole board. The cards are placed facing down before the experiment, and each one carries an identification number on the back (the identification number is used to verify the task number and it will be relevant only for the Primary Investigator). When the subject is ready he/she will give a signal to a computer attached to the digital pin-hole board that the trial has begun by pressing the space bar on the keyboard. The computer will then record the "start" time and will start listening to the digital board for state changes in the inputs.

The subject will then flip the card in front of them and will execute the task indicated on the card. The cards will contain a sequence of instructions of the form "pin X to hole Y" which have to be completed in the stated order. The number of pins to move will vary from card to card, from a minimum of 1 to a maximum of 7. Each pin will have to be relocated only once per trial. The computer will give audio feedback if each step has been accomplished correctly (one "beep" will indicate correct execution, three "beeps" will indicate task completed, while no sounds will be emitted in case of faulty execution). In order to complete the task, the subject will have to reposition the pins in the correct holes. If a pin is misplaced, the subject will not receive audio feedback, and will have to re-attempt their execution until it is correct. The computer will record all insertion times as well as which pin was placed in which hole for later analysis.

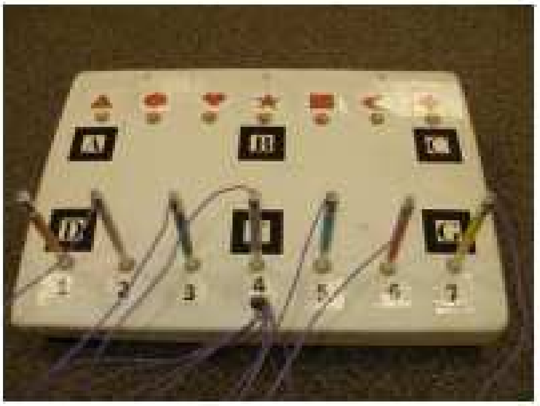
In part B, the subject will be required to execute the same task as in part A with the only difference, being the information feed system in use. The subject will receive the specific task instructions from a see-through HMD system that will overlay an arrow on the digital pin-hole board, indicating which pin to move where. The HMD will be mounted on the subject's head with a snoopy-cap that includes a microphone (not used in this experiment) a pair of headphones that will be used to receive audio feedback from the computer, and a webcam that will be used by the HMD to know where the pin board is and where the pins and holes are. The subject will be tutored on how to wear and remove the system and will practice these procedures until he/she feels comfortable. The subject will be asked whether he/she feels able to remove the system rapidly in case of discomfort. Once the subject is familiarized with the system, the trials will commence as in part A by pressing the "space bar" on the keyboard. The HMD is connected to the computer and will automatically switch through the list of tasks to be accomplished during the trial once each task is successfully completed. All audio feedback is the same as in part A. The HMD will also display how many pins will have to be moved in the trial, and how many have been moved correctly so far by showing a fraction in the upper right corner of the screen (for example, 2/4 will indicate that two out of four pins have been moved). The snoopy cap will be cleaned and disinfected between different subjects with an antibacterial water based spray solution containing 20% Hydrogen Peroxide (H_2O_2).

Ideally, the experiment will be managed such that all subjects execute both part A and part B in the same session, with an even distribution of subjects starting with part A and part B. Before each part, the subjects will complete five trials for familiarization. These trials will be recorded, but will not be used in the later core analysis. The subject can take breaks between trials as desired, and a scheduled 5 minute break will be placed between parts A and B.

At the end of the session, the subject will be asked to fill out a questionnaire on the experiment where he/she will have the opportunity to evaluate it and give written feedback to the Primary Investigator. The questionnaire will be anonymous. Subject responses may be used in the final report. A copy of the questionnaire is attached to this proposal.

<u>Equipment Description:</u>

Digital Pin-Hole Board:



This board was custom made for this experiment and includes:

- 7 identical wired pins (12 cm long, 7 mm in diameter with an electrical contact on the bottom);
- One National Instruments USB-6008 A/D Signal Acquisition Card;
- 14 holes, half of which are connected to the A/D, card while the remaining half are "rest holes" for the pins in their initial position;
- 6 fiducials for HMD visual recognition of the board position and orientation;
- 14 identifiers for pins and holes.

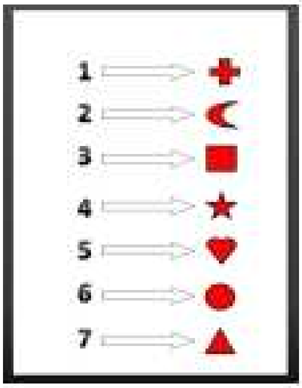
The board dimensions are as follows:

- Length: 38 cm
- Width: 27 cm
- Height: 7 cm (from rest surface)

All corners have been rounded reducing the risk of injuries caused by the board.

Voltages in the circuits are +5 Volts at 500 mA DC at the most. Therefore, there is no significant electric shock risk associate with the board. In any case, the materials used on the board (wood and plastics) isolate the system from the user.
The board is set up in such a way that each pin is set to a different voltage (multiples of 0.15 V). When the pin is in the rest condition, the circuit is open. When a pin is inserted in one of the upper holes which are connected to the A/D card analog input channels, the card will sense a state change. By analyzing this change and the channel on which it occurred, the computer can identify which pin has been inserted in which hole. The card is connected to a PC running a suite of software, Which will record the time at which these changes occur and feed the necessary information to the HMD when in use.

Deck of Cards:
This deck includes 45 paper cards with and identifier number on the back and instructions on the front side. Cards will be pre-arranged before the beginning of the session in a specific order so that they can be interfaced with the computer. Instructions will be displayed in a graphical format showing a pin identifier and a hole identifier connected by a horizontal arrow. The sub-task order will be from top to bottom if more than one subtask is displayed. Card dimensions are as follows:
Height: 10 cm
Width: 8 cm
A sample card is illustrated below:



HMD system:

The HMD is based on a stripped down SONY Glasstron PLM-S700 head mounted display with the following characteristics:

- Resolution = 648x486 (NTSC)
- FOV = 23 sq-deg
- Composite video input
- Complete Headset Approximate Weight: 0.5 Kg

Mounted on the visor is a stripped down General Electric webcam with the following characteristics:

- Resolution: 352x288 (CIF)
- Refresh rate: 30 Hz

The visor assembly is mounted on a "snoopy-cap" with an aluminum mount. All corners have been rounded. Electronics will be covered in order to avoid damage to the equipment due to static charges. Voltages and currents in the circuit do not exceed 9 V at 500 mA DC, therefore the system does not represent an electric shock treat for the user. In any case the system is isolated from the user, being mounted on plastic restraints. The snoopy-cap also includes a pair of headphones and a microphone. The system is adjustable to comfortably fit the subject. The only elements in contact with the subjects' head will be the snoopy-cap soft-goods (ear pads and restraints). The visor will be adjusted in order to be suspended approximately 1 cm from the user's nose and 2 cm from the user's eyes. The visor mount is rigid enough to ensure that there is no contact between the visor assembly and the subject when the user moves his/her head.

Computer:

The computer will be a standard PC or Mac running custom software specifically designed and written for this study. The computer will be in charge of acquiring data from the A/D card, the keyboard and the webcam (if the HMD system is used), and it will store and analyze the data. Interaction between the computer and the subjects will be restricted to audio feedback and the space bar on the keyboard. Any other interaction, including visual feedback from the computer monitor, will not be considered.

4. **Risks and benefits:**

   The experiment will be conducted in a controlled, safe environment. Possible physical discomfort can potentially arise from the use of the HMD. If at any time the subject experiences discomfort, he/she will be encouraged to take the HMD off and decide whether to terminate the experiment. None of the equipment (booklets, digital pin-hole board, HMD and PC) and procedures that will be used in this experiment will present any risk to the subject's health other than the minimal risk described in detail, in the procedures section.

5. **Confidentiality:**

   To help protect confidentiality, the subjects' names will not be included on the collected data. Instead, a code will be placed on the collected data and only the Principal Investigator will be able to link the data to subject identities through the use of an identification key. Only the Principal Investigator will have access to the identification key. All data will be stored on password-protected computers. If a report or article about this research project is written, subject identities will be protected to the maximum extent possible. The data collected in this experiment will be kept indefinitely in the SSL library for future reference.

6. **Information and Consent Forms:**

   Informed consent will be obtained from each subject in writing before the commencement of the experiment. The purpose, background information, detailed procedures, equipment, and risks will be described as above. All of the subjects' questions will be answered. The subjects will then be presented with the consent form and given the opportunity to read it and ask more questions if needed. All subjects will be fluent English speakers and the consent form will not need to be translated into another language.

7. **Conflict of Interest:**

   This research is not conducted in collaboration with the private sector. No party has financial or employment interest in the outcome of this research. There are no conflicts of interest.

8. **HIPAA Compliance:**

   No HIPAA protected information will be used. No information will be obtained from health care providers.

9. **Research Outside of the United States:** N/A

10. **Research Involving Prisoners:** N/A

CONSENT FORM

Project Title	***Augmented Reality for Space Application: Performance and Accuracy Impact on Simple Unrehearsed Tasks. Comparison Between In and Out-Of-Field-Of-View Information Feed Methods.***
Why is this research being done?	*This is a research project being conducted by Dr. David Akin and Massimiliano Di Capua of the Space Systems Laboratory at the University of Maryland, College Park. The purpose of this research is to quantify and analyze the effect on performance and accuracy in-field-of-view information displays when attempting unrehearsed tasks. The main goal is to find evidence of important benefits arising from the use of this system that could justify its intrinsic complexity and prepare the way for a future implementation in spacesuit applications.*
What will I be asked to do?	*In this experiment, you will be asked to perform a simple task as quickly and as accurately as you can. Your goal is to extract a specific pin and insert it back in a specific hole on the board in front of you. You will receive the necessary information through two different systems: A set of cards and a Head Mounted Display (HMD)(Approximate complete headset weight: 0.5 Kg) .* *The test will last approximately 1.5 hours and will consist of 40 trials for each information display system. With each system, you will have 5 familiarization trials, after which the experiment will start. In order to complete each trial you shall:* *- Initiate the task by pressing the number indicated on the back of the card, followed by the enter key on the keyboard by your side;* *- Flip the information card or look through the visor for task information;* *- Execute the task* *Once the task has been completed (all instructed pins have been relocated), the computer will give you an audio signal. Possible signals are as follows:* *-Single beep: Pin inserted and recognized.* *-Three beeps: Task completed.* *- No sound: Task not completed, one or more pins are misplaced.* ***In case you have misplaced a pin, you will have to verify and correct your final configuration until you hear the three beeps.*** *Information Cards:* *Once you have pressed the spacebar, you can flip the card in front of you. On the card you will find numbers and symbols representing the pins and holes on the board. Proceed in the task by moving the pin with the same number as shown on the card preceding the arrow, and placing it in the hole with the symbol shown after the arrow. Once the first row on the card has been executed, proceed to the second row and so on.*

	HMD: *Once you have adjusted the HMD properly and have a comfortable fit, you can press the space bar on the keyboard by your side and commence the trial. Once the trial has started, look at the board through the visor, and you should see an arrow starting from the base of the pin that you will have to extract and pointing to the hole you should insert it in. On the board, you will notice there are 6 black squares with letters inside. Those are the fiducials used by the HMD. The HMD will look for those through the camera mounted on it in order to determine where to display the arrow. If you do not see an arrow on the board when looking through the visor, make sure that at least one of the fiducials is not obstructed, and center it in your field of view.* *In both cases, once the trial has begun, a computer will record how long it took you to complete the task.* *If at any time you should experience fatigue or any kind of discomfort, take a break! You will be given a 5 minute break once you have completed 40 trials. You can take as many breaks as you wish in between trials.* *You are also allowed to reposition the equipment in front of you (keyboard and/or board) before each trial if you wish. Those changes will be recorded by taking a picture of the setup. In this case you will be asked to step out of the camera field of view to protect your identity. At the end of the experiment, you will be asked to complete a short questionnaire about basic personal information, vision condition, and your feedback on the experiment.*
What about confidentiality?	*Your personal information will be confidential. To help protect your confidentiality: (1) your name will not be included on the collected data; (2) a code will be placed on the collected data; (3) through the use of an identification key, the researchers will be able to link your data to your identity; (4) only the researchers will have access to the identification key; and (5) all data will be password-protected. If we write a report or article about this research project, your identity will be protected to the maximum extent possible. The data collected in this experiment will be kept indefinitely in the SSL library for future reference.* *Your information may be shared with representatives of the University of Maryland, College Park or governmental authorities if you or someone else is in danger or if we are required to do so by law.*
What are the risks of this research?	*There is no added risk to your person due to the equipment used in this experiment. Although mild, side effects could occur and include:* - *Eye fatigue* - *Dizziness* - *Discomfort* ***If you should experience any of the above, please notify the investigator and interrupt the experiment immediately. In the case you are wearing the HMD, please remove it. You can continue, if desired, after the symptoms have gone.***

What are the benefits of this research?	*This research is not designed to benefit you personally, but it will give us more information for the future development of this system and eventually may lead to its integration in a future space exploration program.*
Do I have to be in this research? May I stop participating at any time?	*Your participation in this research is completely voluntary. You may choose not to take part at all. If you decide to participate in this research, you may stop participating at any time. If you decide not to participate in this study or if you stop participating at any time, you will not be penalized or lose any benefits to which you otherwise qualify.*
Is any medical treatment available if I am injured?	*The University of Maryland does not provide any medical, hospitalization or other insurance for participants in this research study, nor will the University of Maryland provide any medical treatment or compensation for any injury sustained as a result of participation in this research study, except as required by law.*
What if I have questions?	*This research is being conducted by Dr. David Akin and Massimiliano Di Capua of the Department of Aerospace Engineering at the University of Maryland, College Park. If you have any questions about the research study itself, please contact:* **Dr. David Akin** **University of Maryland** **Building 382 Room 2100D** **College Park, MD 20742** **(e-mail) dakin@ssl.umd.edu** **(telephone) 301-405-1138** *or* **Massimiliano Di Capua** **University of Maryland** **Building 382 Room 1100C** **College Park, MD 20742** **(e-mail) maxdicapua@yahoo.com** **(telephone) 202-489-2626** *If you have questions about your rights as a research subject or wish to report a research-related injury, please contact:* **Institutional Review Board Office** **University of Maryland** **College Park, MD 20742** **(e-mail) irb@deans.umd.edu** **(telephone) 301-405-0678** *This research has been reviewed according to the University of Maryland, College Park IRB procedures for research involving human subjects.*

Statement of Age of Subject and Consent	*Your signature indicates that:* • *you are at least 18 years of age;* • *the research has been explained to you;* • *your questions have been fully answered; and* • *you freely and voluntarily choose to participate in this research project.*	
Signature and Date	**NAME OF SUBJECT**	
	SIGNATURE OF SUBJECT	
	DATE	

1 9 2009

Augmented Reality for Space Application: Performance and Accuracy Impact on Simple Unrehearsed Tasks. Comparison Between In and Out-Of-Field-Of-View Information Feed Methods.

Subject Pre-Questionnaire

1. Age: __

2. Gender: __ male __ female

3. What is your affiliation with the University College of Engineering?

__ undergraduate student __ faculty

__ graduate student __ staff

4. Are you left or right-handed?

__ left __right __no preference

5. Do you have any physical impairments? (select all that apply)

__ reduced vision, if so is it corrected with glasses or lenses?

__yes __no

if no, is it likely to affect your speed and accuracy in movements?

__yes __no

__ reduced arm mobility,

if so is it likely to affect your speed and accuracy in movements?

__yes __no

6. Do you have previous experience with HMD devices?

__yes __no

Augmented Reality for Space Applications: Performance and Accuracy Impact On Simple Unrehearsed Tasks. Comparison between In and Out-Of-Field-Of-View Information Feed Methods.

Subject Questionnaire

Basic Information:

Sex: ____________ Age: ____________

(circle appropriate response)

Do you have reduced vision?	yes	no
If yes, do you wear contact lenses?	yes	no
Do you wear glasses?	yes	no
Are you color blind?	yes	no
Did you experience any side effects during the trials?	yes	no

If you answered "Yes" to the above, please list the symptoms:

____

____

Response to Experiment: (check appropriate box: 1=not at all, 5=very)

Cards:

Question	1	2	3	4	5
Were the trials easy?					
Were the cards hard to read?					
Did you have any difficulty distinguishing the pins?					
Did you have any difficulty distinguishing the holes?					
Were the feedback tones clear?					
Were you satisfied with your performance?					

HMD

Question	1	2	3	4	5
Were the trials easy?					
Was the HMD hard to use?					
Did you have any difficulty distinguishing the pins?					
Did you have any difficulty distinguishing the holes?					
Was the HMD interface intuitive?					
Did the HMD obstruct your view too much?					
Was the HMD comfortable?					
Did you experience any side effects?					
Were you able to distinguish colors on the HMD?					
Were sound warnings clear?					
Were you satisfied with your performance?					

Overall, which system did you prefer? **Paper Cards** **HMD**

What changes would you like to see in new generation HMDs?

____

____

Additional Comments:

Recruitment Email:

To: ssl@ssl.umd.edu

From: maxdicapua@yahoo.com

Subject: Volunteers needed for Augmented Reality study!

Dear all,

As many of you know, I'm ready to start testing for my research!

The experiment is called **Augmented Reality for Space Applications: Performance and Accuracy Impact On Simple Unrehearsed Tasks. Comparison between In and Out-Of-Field-Of-View Information Feed Methods.**

In this experiment, we will simulate a simple unrehearsed task by removing a specific peg and inserting it in a different hole on my brand new digital pin-hole board. You will not have any prior knowledge of the sequences to be executed and you will receive all necessary information just before commencing the task. The experiment will be divided in two parts differing on the information display method you will be using. In the first part you will receive a deck of cards with all the instructions, while in the second you will wear a head mounted display that will overlay on the pin-hole board a schematic of the task. I'm looking for 10-15 subjects, at least 18 years of age, without visual impairment (glasses or contact lenses are fine!) or significantly reduced arm mobility. The experiment should last approximately one hour and a half. Participation is completely voluntary. No compensation will be awarded.

I could use all the help I can get, therefore, if you are eligible and interested, please let me know!

Thank you!!!!!!!!!!

Massimiliano

Massimiliano Di Capua
Graduate Student
Department of Aerospace Engineering
University of Maryland
Email: maxdicapua@yahoo.com
Phone: (202) 489-2626

Bibliography

[1] R. T. Azuma, *A Survey of Augmented Reality 6*. Teleoperators and Virtual Environments, 4 August 1997,p. 355–385.

[2] R. T. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, B. MacIntyre, *Recent Advances in Augmented Reality 6*. IEEE, November/December 2001, 0272-1716/01, p. 34–47.

[3] H. Kato, M. Billinghurst, *Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System*. IWAR, 1999.

[4] Y. Ohta, H. Tamura, *Mixed Reality: Merging Real and Virtual Worlds*. Springer–Verlag New York Inc., 1999, ISBN 3–540–6562355.

[5] M. Haller, M. Billinghurst, B. Thomas, *Merging Technologies of Augmented Reality: Interfaces and Design*. IDEA Group Publishing, 2007, ISBN 1–59904–066–2.

[6] M. Agan, A. Voisinet, A. Deveraux, *NASA's Wireless Augmented Reality Prototype (WARP)*. Jet Propulsion Laboratory, January 1999.

[7] C. E. Carr, S. J. Schwartz, I. Rosenberg, *A Wearable Computer for Support of Astronaut Extravehicular Activity*. IEEE, 2002, Proceedings of the 6th International Symposium on Wearable Computers (ISWC.02) 0-7695-1816-8/02.

[8] J. Marmolejo, *Use of Voice Commands to Provide a Hands Free Information System for Space Station Extravehicular Activity*, April 1987, Proceedings of Speech Technology.

[9] J. Marmolejo, C.G. Gernux, R.W. Blaser, *A Helmet Mounted Display Demonstration Unit for a Space Station Application*. July 1989, 19th Intersociety Conference on Environmental Systems.

[10] J. Marmolejo, *Helmet-Mounted Display and Associated Research Activities Recently Conducted by the NASA/Johnson Space Center*. SPIE, April 1994, Proceedings of International Symposium on Optical Engineering in Aerospace Sensing.

[11] J. Marmolejo, *An Electronic Cuff Checklist Information Display for Extravehicular Activity*. 1996, 26th International Conference on Environmental Systems.

[12] M.F. Stollen, B. Dillow, S. Jacobs, D.L. Akin, *Interface for EVA Human-Machine Interaction*. 2008, SAE, 08ICES-0166.

[13] E. Hodgson, R. Sidgreaves, *Requirements and Potential for Enhanced EVA Information Interfaces*. 2003, SAE, 2003-01-2413.

[14] E. M. Jones, *Apollo Lunar Surface Journal*. July 2000, (http://www.hq.nasa.gov/office/pao/History/alsj/).

[15] C.E. Carr, D.J. Newman, *Supporting Martian Field Geology* 2001, Geological Society of America.

[16] D. L. Akin ,*Robotic Augmentation of EVA for Hubble Space Telescope Servicing* 2003, AIAA Space 2003 Conference and Exposition, AIAA-2003-6274.

[17] R.O. Ambrose ,*Robonaut: NASAs Space Humanoid* 2000, IEEE Intelligent Systems, p. 57-63.

[18] J. Forster, E.T. Higgins, A.T. Biancoc*Speed/accuracy decisions in task performance: Built-in trade-off or separate strategic concerns?* 2003, OBHDP.