NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DEVELOPING ARTICULATED HUMAN MODELS FROM LASER SCAN DATA FOR USE AS AVATARS IN REAL-TIME NETWORKED VIRTUAL ENVIRONMENTS

by

James Allen Dutton

September 2001

Thesis Advisor:	Eric R. Bachmann
Thesis Co-Advisor:	Xiaoping Yun

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2001	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Title (Mix case letters) Developing Articulated Human Models From Laser Scan Data For Use as Avatars in Real-Time Networked Virtual Environments	**5. FUNDING NUMBERS**
6. AUTHOR(S) James Allen Dutton	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

With the continuing gain in computing power, bandwidth, and Internet popularity there is a growing interest in Internet communities. To participate in these communities, people need virtual representations of their bodies, called avatars. Creation and rendering of realistic personalized avatars for use as virtual body representations is often too complex for real-time applications such as networked virtual environments (VE). Virtual Environment (VE) designers have had to settle for unbelievable, simplistic avatars and constrain avatar motion to a few discrete positions.

The approach taken in this thesis is to use a full-body laser-scanning process to capture human body surface anatomical information accurate to the scale of millimeters. Using this 3D data, virtual representations of the original human model can be simplified, constructed and placed in a networked virtual environment.

The result of this work is to provide photo realistic avatars that are efficiently rendered in real-time networked virtual environments. The avatar is built in the Virtual Reality Modeling Language (VRML). Avatar motion can be controlled either with scripted behaviors using the H-Anim specification or via wireless body tracking sensors developed at the Naval Postgraduate School. Live 3D visualization of animated humanoids is viewed in freely available web browsers.

14. SUBJECT TERMS Virtual Environments, Laser Scanning, VRML, Humanoid Animation 1.1 Specification, 3D, virtual humans, avatars, Java, DIS-Java-VRML, Web3D Consortium, motion trackers	**15. NUMBER OF PAGES** 53
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

DEVELOPING ARTICULATED HUMAN MODELS FROM LASER SCAN DATA FOR USE AS AVATARS IN REAL-TIME NETWORKED VIRTUAL ENVIRONMENTS

James Allen Dutton
Lieutenant, United States Navy
B.S., Oregon State University, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION

from the

NAVAL POSTGRADUATE SCHOOL
September 2001

Author: James A. Dutton
James A. Dutton

Approved by: Eric Bachmann
Eric Bachmann, Thesis Advisor

Xiaoping Yun, Thesis Co-Advisor

Rudy Darken, Academic Associate
Modeling, Virtual Environments, and Simulation Academic Group

Michael Zyda, Chair
Modeling, Virtual Environments, and Simulation Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

With the continuing gain in computing power, bandwidth, and Internet popularity there is a growing interest in Internet communities. To participate in these communities, people need virtual representations of their bodies, called avatars. Creation and rendering of realistic personalized avatars for use as virtual body representations is often too complex for real-time applications such as networked virtual environments (VE). Virtual Environment (VE) designers have had to settle for unbelievable, simplistic avatars and constrain avatar motion to a few discrete positions.

The approach taken in this thesis is to use a full-body laser-scanning process to capture human body surface anatomical information accurate to the scale of millimeters. Using this 3D data, virtual representations of the original human model can be simplified, constructed and placed in a networked virtual environment.

The result of this work is to provide photo realistic avatars that are efficiently rendered in real-time networked virtual environments. The avatar is built in the Virtual Reality Modeling Language (VRML). Avatar motion can be controlled either with scripted behaviors using the H-Anim specification or via wireless body tracking sensors developed at the Naval Postgraduate School. Live 3D visualization of animated humanoids is viewed in freely available web browsers.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
A. BACKGROUND 1
B. MOTIVATION 2
1. Entertainment – A Presence and Analysis Tool 2
2. Collaborative Meetings – Its All in the Body Language 3
3. DoD Areas of Interest – Cutting-Edge War Fighting and Training 3
C. OBJECTIVES 4
D. THESIS OUTLINE 5

II. PROBLEM STATEMENT 7
A. INTRODUCTION 7
B. PROBLEM STATEMENT 7
C. PROPOSED SOLUTION 8
D. RESEARCH FOCUS 8
E. DESIGN CONSIDERATIONS 8
F. SUMMARY 9

III. 3D SCANNING OF HUMANS 11
A. INTRODUCTION 11
B. METHODS FOR 3D SCANNING OF HUMANS 11
1. Stereoscopic Vision Scanners 11
2. Moire Projection Scanners 12
3. Time-of-Flight (TOF) Scanners 13
4. Laser Triangulation Scanners 14
C. THE CYBERWARE WHOLE BODY SCANNING PROCESS 15
1. System Background 15
2. System Operation 15
D. SUMMARY 16

IV. RELATED WORK 17
A. INTRODUCTION 17
B. VIRTUAL REALITY MODELING LANGUAGE (VRML) 17
C. VRML AND JAVA WORKING TOGETHER 19
D. THE HUMAN ANIMATION 1.1 SPECIFICATION 21
E. NANCY: THE H-ANIM 1.1 STANDARD 24
F. HUMAN MOTION TRACKING TECHNOLOGIES 26
1. Mechanical Trackers 26
2. Magnetic Trackers 27
3. Optical Trackers 28
4. Acoustic Trackers 28
5. Inertial and Magnetic Tracking 29
G. SUMMARY 30

V. INITIAL DEVELOPMENT EFFORTS 31
A. INTRODUCTION...31
B. HANDLING LASER SCAN DATA 31
1. Reducing Laser Scan Output Complexity 31
2. Translating Between Files...33
3. Segmenting The Avatar...34
C. CONSTRUCTING THE AVATAR 36
D. USING JAVA TO PROVIDE REAL-TIME NETWORKED CONTROL 39
E. SUMMARY..41

VI. CONCLUSIONS AND RECOMMENDATIONS...43
A. GENERAL THESIS CONCLUSIONS 43
B. SPECIFIC CONCLUSIONS AND RESULTS...43
1. Constructing Anatomically Accurate Avatars 43
2. Flexible Avatar Control...43
3. Source Code is Platform-Independent and Open-Source 43
4. Simple 3D Authoring is Combined With Powerful Programming Capability..44
C. LESSONS LEARNED 44
D. RECOMMENDATIONS FOR FUTURE WORK.....................................44
1. Automating Avatar Segmentation and Construction.....................44
2. Updating File Translators to Retain Texture Information............45
3. Keep Joints Connected in All Positions 45
4. Increase Behavior and Motion Libraries..45
5. Update Existing Body-Tracking Code 46
6. Construct Avatar in Other Programming Languages 46
E. SUMMARY..46

LIST OF REFERENCES 47

INITIAL DISTRIBUTION LIST..51

LIST OF FIGURES

Figure 1. Simple Opt ical Triangulation. 11
Figure 2. Moire Light Patterns....12
Figure 3. Time-of-Flight Laser Scanning...13
Figure 4. Laser Triangulation Scanning....14
Figure 5. Cyberware Scanner Model WB4. From Ref. [5]. 15
Figure 6. Example of Simple VRML Source Code and Output. From Ref. [17]...........18
Figure 7. Script Node Interface. From Ref. [19]...19
Figure 8. Example Script Node and Java Interaction. From Ref. [17]...........................21
Figure 9. H-Anim Spec 1.1 Hierarchy. From Ref. [20]. 23
Figure 10. Nancy Demonstrating the Stand Behavior...24
Figure 11. Nancy Demonstrating the Walk Behavior....25
Figure 12. Nancy Demonstrating the Run Behavior....25
Figure 13. Nancy Demonstrating the Jump Behavior. 26
Figure 14. Initial Laser Scan Output of the Author (150,000 Polygons). 32
Figure 15. Translated VRML Avatar (10,000 Polygons). 34
Figure 16. Reduced Laser Scan Imported Into Maya...36
Figure 17. Initial VRML/H-Anim 1.1 Avatar From Laser-Scan (Untextured). 37
Figure 18. Texture Mapped, Articulated Avatar Capable of H-Anim 1.1 Scripted Movement. 39

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to acknowledge the open-source distribution material from the Web3D Consortium, in particular, the DIS-Java-VRML and Human Animation Working Groups, which made this thesis research possible.

I would like to thank Cindy Ballreich for allowing the use of her model Nancy. I would also like to thank my advisors, Dr. Eric Bachmann and Dr. Xiaoping Yun for their technical expertise, support, focus and friendship throughout this project. A special thanks to Dr. Don Brutzman for his invaluable assistance with VRML, and to Don McGregor for sharing his extensive Java networking knowledge.

Last but not least, I would like to thank my family (Kim, Jennifer and Michelle) for their love and infinite patience. Without their loving support this work could not have been completed.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

In 1965 Gordon Moore, a cofounder of Intel Corporation, predicted that computing power would roughly double every 18 to 24 months [Ref. 1]. This prediction, known as "Moore's Law", has been remarkably accurate for nearly forty years, and there is every indication that it will continue to do so for the foreseeable future. Computer graphical power, however, has been surpassing even Moore's Law, approximately *cubing* every 18 to 24 months [Ref. 2]. Internet bandwidth is experiencing an evolution as well, with ADSL/DSL (Asynchronous Digital Subscriber Line/Digital Subscriber Line) and cable modems becoming more commonplace in homes and offices across America. At the end of 2000, there were over two million American homes with DSL/ADSL and the growth rate is accelerating [Ref. 3].

With these computing power and Internet bandwidth gains, and the increasing popularity of the worldwide web, there is a growing interest in networked virtual environments (NVE). A NVE is a software environment in which multiple users interact with each other in real-time, even though these users may not be physically in the same room, or even on the same continent. These environments could consist of anywhere from two to thousands of people, possibly all interacting in the same environment. Some examples are business conferences, engineering design forums, entertainment applications, distance learning and Department of Defense (DoD) simulations. [Ref. 4]

When entering a NVE, each participant assumes a virtual persona, visually represented by an *avatar*, which includes a body structure model, motion model, physical model, and possibly many other characteristics depending on the application [Ref. 4]. This thesis is an attempt to construct an articulated, anatomically accurate avatar and place it within a NVE, which may be viewed via freely available web browsers. The avatar is the result of a full-body laser-scanning process and is accurate to a scale of millimeters [Ref. 5]. Consideration has been given to graphical complexity and bandwidth requirements, with the final model being extremely efficient and usable on today's computers. The avatar can be used in virtual applications. Avatar movement can be controlled via pre-scripted movements, such as the VRML H-Anim specification [Ref. 6], or made to shadow the controlling person's movement via real-time motion capture [Ref. 7].

B. MOTIVATION

The motivation for this project is to provide virtual environment (VE) designers with the technology for "photo-realistic" avatars. These avatars are a replica of their human source and are extremely lifelike since their movements are driven by either pre-scripted human animation data or actual human input, and are scaled to the user to which the motion data pertains. Three examples of possible applications are entertainment, collaborative meetings, and DoD areas of interest. Specific details regarding possible scenarios are now examined.

1. Entertainment – A Presence and Analysis Tool

Presence is defined as the feeling of "being there." Imagine a home video game system where the input controller is no longer a joystick, keypad, mouse or other artificial device, but is instead the body of the gamer. The movement of the user's physical body is translated into a digital representation by a motion tracking system. This representation is then connected to their on-screen avatar, which will then mimic every movement of the user. The result is that the user moves their body in the real world in exactly the same manner they want their virtual alter ego to move. For example, in a fighting game, the participant would be executing the moves in the real world, with the avatar mimicking their every movement and the virtual environment then responding appropriately. One can easily imagine such an interface for several different genres, from role-playing to virtual sports.

Also in the realm of entertainment, but on a slightly different tack is performance analysis. Consider the example of virtual golf. A user who is motion tracked can actually swing the golf club as they do in real life to play the game. Not only can they be entertained from a gaming perspective, but they can also be instructed. The application could monitor their swing, and point out weaknesses and areas of improvement. They could redo the shot under the exact same conditions, experiment with their swing, and observe the outcome. Also, their swing could be recorded and played back for useful self-analysis.

2. Collaborative Meetings – Its All in the Body Language

The world community has long recognized the need for face-to-face meetings for effective communication. Since a large part of the way humans communicate is via body language, interacting through text alone is not sufficient in many cases. Subtle nuances of behavior that could be vitally important can be missed without certain non-verbal cues.

Streaming video has been used as one solution to this problem but remains an expensive solution, both in terms of bandwidth and computer hardware. Such traditional media is also limited in that the viewpoint is fixed. The viewer can only see the action from the angle it was recorded, and is helpless to view the scene from another angle if circumstance or preference dictate otherwise. Physical constraints may make it impossible or impractical to place cameras at certain vantage points.

Avatars offer a more flexible alternative, with lower bandwidth requirements. Active participants are tracked, with their avatars mimicking their behavior in the virtual world. Thus, all of the participants can see a shrug, a shake of the head, or a hand placed to the chin in thought. Flexibility is provided by complete virtual camera control. All participants can place their viewpoint anywhere they wish and zoom in or out. Engineering and architectural design collaborative sessions, distance learning and business meetings are examples of this type of application.

3. DoD Areas of Interest – Cutting-Edge War Fighting and Training

The DoD has long been the largest developer of large networked virtual environments, with the goal of training personnel more effectively and economically [Ref. 4]. Simulator Networking (SIMNET) [Ref. 8] and the Distributed Interactive Simulation (DIS) protocol [Ref. 9] are examples of DoD interest in this area. To date, the use of human entities or dismounted infantry (DI) has been limited in most high-resolution virtual simulations [Ref. 10].

With the use of realistic avatars, the military simulation role may be expanded to include action at the individual soldier level, vice incorporating only large-scale troop movements. Networked virtual rehearsal becomes possible, thereby eliminating geographical separation difficulties between command and troops. Rehearsal can be done with less bandwidth and more securely than current methods.

Currently, physical descriptions with accompanying distinguishing physical feature descriptions are part of every military service record. This information could also include laser scan data. Such data would describe their appearance as well as their physical dimensions down to the millimeter. Besides being useful as a means of accurate identification, this data would also be available for use in creating personalized avatars. The scan output could be called up to render every member in 3D. One possible application would be to drive their avatars with motion-tracking sensors. In this manner, commanders may view a battlefield simulation as it unfolds, with their view unlimited by physical constraints, either taking a "gods-eye" view or zooming in to one specific combatant according to their preference. The mission may also be recorded and played back for debriefing, and the playback camera position would not be limited to the original position when the data was recorded, thus giving a distinct advantage over current conventional recording and playback methods.

C. OBJECTIVES

The objective of this research is to construct human replica avatars for use in virtual environments using data obtained from a whole body laser scanning process. To achieve this objective, the following areas are addressed:

- The complexity of laser scan data must be reduced so that the avatars may be rendered efficiently with current computing technology.
- The data must be translated into one or more universal formats that are platform independent and will therefore run under several different operating

systems. Optimally, the chosen file format will have open source code that is freely distributed.

- The data obtained from the scanning process is a "data soup" in that it is a single figure with no segmentation. The output data must be organized to segment the body in order to provide for full articulation and realistic movement.
- The avatar must be built from its body segment building blocks, be physically accurate and visually compelling.
- Avatar movement must be possible through scripted (pre-defined) motion, and also through real-time input, such as over a network from motion trackers.

D. THESIS OUTLINE

This chapter describes the background, motivation and objectives to be achieved in order to produce 3D avatar replicas from laser scan data. Chapter II contains a concise problem statement for this thesis.provides an overview of 3D human scanning technologies with an in-depth look at the 3D laser triangulation scanning method chosen for this research. Chapter III provides an overview of 3D human scanning technologies with an in-depth look at the 3D laser triangulation scanning method chosen for this research. Chapter IV discusses the Virtual Reality Modeling Language (VRML), how VRML and Java work together, humanoid animation and human motion tracking. Chapter V discusses initial development efforts to include scan complexity and file format issues, organizing laser scan data into body segments, selection of a 3D rendering engine, scripted avatar behaviors, and communicating real-time motion tracking input over networks. Chapter VI provides thesis conclusions and recommendations for future or follow-on research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. PROBLEM STATEMENT

A. INTRODUCTION

This chapter defines the problem examined for this thesis and offers a proposed solution. Further, the focus of this research is discussed, and design issues that were considered during model implementation are addressed.

B. PROBLEM STATEMENT

With the explosion of the worldwide web, people from all over the world are interacting electronically with each other in ever-increasing numbers [Ref. 3]. Networked virtual environments (NVEs) provide one form of electronic interaction among humans. A NVE is a software environment in which multiple users interact with each other in real-time, even though these users may not be physically in the same room, or even the same continent [Ref. 4].

When entering a NVE, each participant assumes a virtual persona, called an *avatar,* which includes a graphical representation, body structure model, motion model, physical model, and possibly many other characteristics depending on the application [Ref. 4]. While the film industry has enjoyed much success digitizing humans, much processing time is required to create highly complex models that are unable to be rendered over networks with real-time interaction. Past solutions that have allowed real-time interaction have compromised on avatar quality, resulting in overly simplified models that reduce virtual reality effectiveness by decreasing the user's sense of presence. Virtual environment applications that require exact dimensions of the human body may also suffer, as simplistic avatars often bear little resemblance to the original model in both appearance and measurement. Poorly sized models can result in lessening the user's sense of presence, since the avatar's limbs may appear to go through virtual objects, including the avatar itself. Manual exercises, such as reaching out and manipulating an object become difficult if avatar dimensions do not equal the controlling human's dimensions.

C. PROPOSED SOLUTION

The proposed solution for these challenges is to develop a high-resolution, dimensionally accurate human model, or avatar with a realistic appearance. The model must be efficient enough to run easily on today's computers, and scale well so that many avatars could be rendered simultaneously while maintaining a satisfactory frame rate. Further, avatar control through either pre-scripted actions or real-time updates via networking must be supported. Finally, the system must be platform independent to permit hardware and software flexibility.

D. RESEARCH FOCUS

The focus of this research is to build a fully articulated human model from laser scan data for use as an avatar. The model must be simplified, and then built using an international standard for networked humanoid animation, Humanoid Animation Specification 1.1 (H-Anim 1.1). The H-Anim 1.1 canonical exemplar Nancy.wrl, written in Virtual Reality Modeling Language (VRML), is used as an avatar foundation. Using H-Anim 1.1 and VRML provides the capacity for pre-scripted avatar control. Additionally, Java and VRML must be made to efficiently work together to provide the capability of real-time networked avatar control. Using Java and VRML ensures platform independence. The implementation must be able to accept quaternion inputs to be compatible with the Magnetic Angular Rate Gravity (MARG) motion tracking sensors developed at the Naval Postgraduate School.

E. DESIGN CONSIDERATIONS

The most significant design consideration is how to transform the "data cloud" obtained from laser scans into a fully articulated, segmented avatar. Multiple proprietary and non-proprietary data conversion methods must be examined before an implementation is chosen. The selected implementation first uses Cyberware Laboratories "Decimate" software package to reduce model complexity and provide file format translations, then uses "Maya" from Alias/Wavefront for avatar segmentation.

Other major design considerations are: choosing which scanning method to use to capture body surface information; constructing the avatar using Nancy.wrl; implementing networking capability via DIS-Java-VRML; and providing for quaternion input for networked control.

F. SUMMARY

This chapter defines the problem addressed by this research and offers a proposed solution. The focus of this thesis is discussed, and design considerations are examined.

THIS PAGE INTENTIONALLY LEFT BLANK

III. 3D SCANNING OF HUMANS

A. INTRODUCTION

This chapter provides an overview of the various 3D scanning technologies available at the time of the writing of this thesis. An in-depth discussion of human laser scanning follows, as it is the 3D scanning method chosen for this research.

B. METHODS FOR 3D SCANNING OF HUMANS

Although various 3D scanning methods have been available for the last two decades, recent advances in image sensing have increased their speed and accuracy tremendously [Ref. 11]. An overview of current human 3D scanning technologies follows.

1. Stereoscopic Vision Scanners

Stereoscopic scanning is a passive optical technique. Two or more digital images are taken from known locations. These images are then processed to find correlations between objects in the images. Figure 1 illustrates the principles of stereoscopic vision. Points a, b and c represent common features seen from two separate viewpoints.

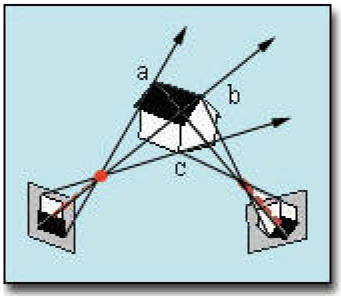


Figure 1. Simple Optical Triangulation.

Each viewpoint has both a focal distance and angle. The two different distance/angle combinations are used to calculate the distance to the common elements. Human binocular vision works in the same manner. The further away the common elements, the more the two separate focal distance/angle pairs will agree. As the object moves closer relative to the viewer, focal angles between the two vision sensors vary increasingly, making possible a distance estimate.

In theory, this method requires very little hardware: at the minimum just two cameras and a computer to process the images. In reality, however, correlations between images are often difficult to ascertain, necessitating the use of special light projectors that add to both system complexity and cost. Furthermore, with current computing power processing the photographic images can take ten minutes or more, and the resulting image quality is still vastly inferior to that obtained with laser-based systems. Image quality can be improved through the use of higher-resolution cameras, but at the expense of considerably longer processing times and hardware costs. [Ref. 12]

2. Moire Projection Scanners

In Moire projection scanning, a series of structured light patterns are projected onto the object to be digitized. The shape of the object causes the base pattern to be distorted from its original design. By analyzing this distorted light pattern the shape of the object is calculated, and x-y-z coordinates are then produced. Figure 2 below is an example of a typical Moire pattern on an object. [Ref. 13]

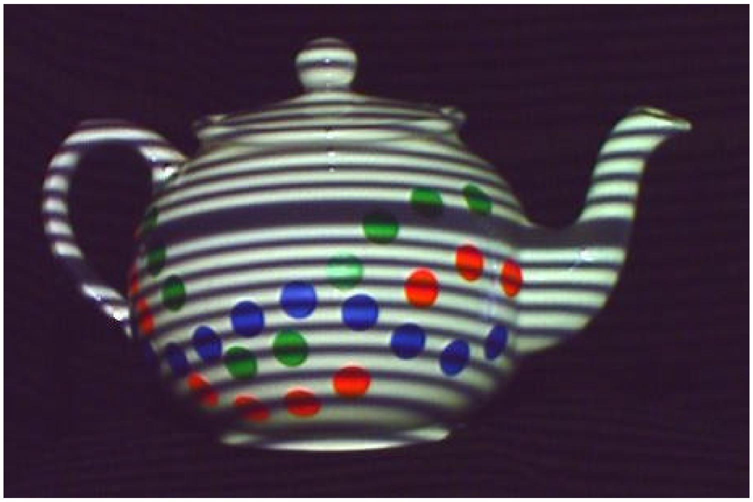


Figure 2. Moire Light Patterns.

Moire scanning has several disadvantages. The object must be lit solely by the capturing light source, as bright ambient light disrupts the contrast of the pattern and interferes with the resolution of the scan [Ref. 13]. Also, scan quality depends on the color of the object, possibly resulting in large scan errors and reduced resolution. Although some of the best Moire scanners can offer fidelity comparable to laser scanning, scan time is greatly extended and the units are more costly than their laser counterparts [Ref. 12].

3. Time-of-Flight (TOF) Scanners

These scanners use a type of laser scanning based on the technique of Laser Imaging Detection and Ranging (LIDAR). Distance is measured by comparing the phase of the returned laser beam to the original, allowing for very accurate scanning of very large objects. Figure 3 graphically depicts the principles of time-of-flight scanning. [Ref. 14]

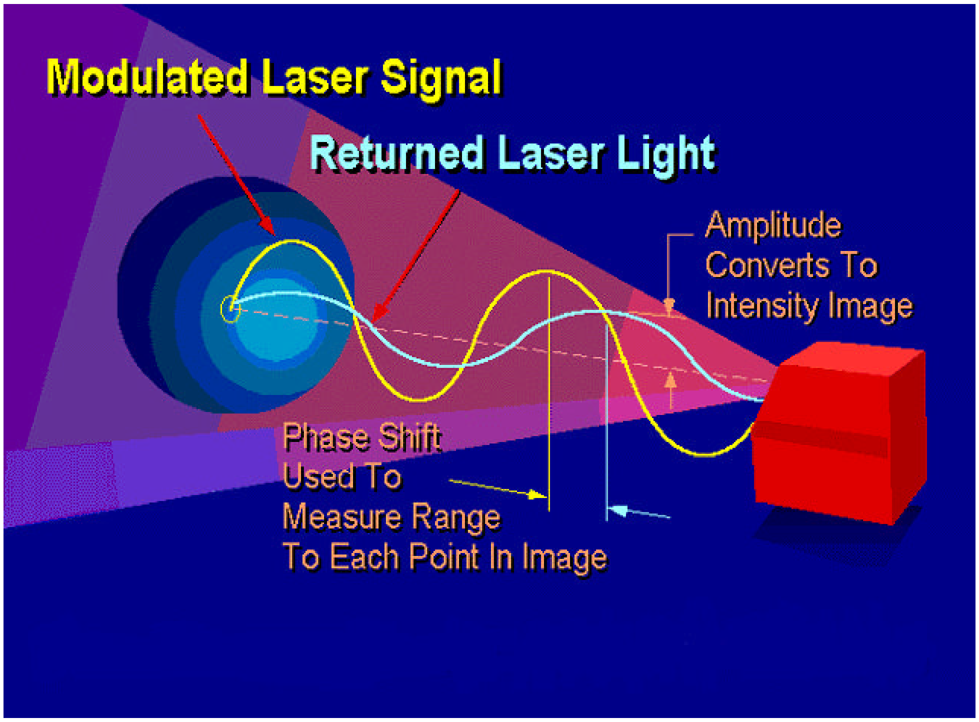

Modulated Laser Signal
Returned Laser Light
Amplitude Converts To Intensity Image
Phase Shift Used To Measure Range To Each Point In Image

Figure 3. Time-of-Flight Laser Scanning.

Unlike triangulation scanning, TOF scanning accuracy remains constant regardless of the distance from the scanner. This makes TOF scanners ideal for large objects. Unfortunately, only geometry information is captured and the object's texture information is not acquired. Additionally, TOF scanning time is much slower than laser triangulation. With current technology, TOF scanning is not practical for the fast digitization of small and medium-size objects. [Ref. 12]

4. Laser Triangulation Scanners

Laser triangulation is a stereoscopic technique that calculates distances to an object by means of a video camera and a laser light source. Figure 4 is a schematic representation of the laser triangulation process. A laser beam is reflected from a mirror onto the object to be scanned. The laser light is scattered by the object and is picked up by the detector, in this case a video camera. Since the triangulation distance and transmitted light angle are known, the distance to the object may be calculated from the received image using basic trigonometry.

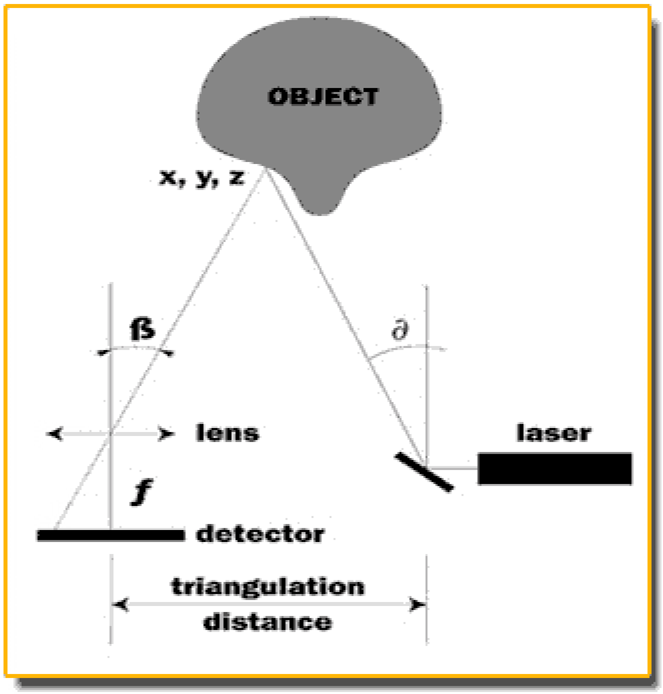


Figure 4. Laser Triangulation Scanning.

This method is fast when compared with other scanning methods. Whole body scan times are on the order of seconds, and are accurate down to the millimeter. Texture capture is also possible, allowing for highly detailed scan output. The end product is a lifelike, believable digital representation of the original object.

C. THE CYBERWARE WHOLE BODY SCANNING PROCESS

1. System Background

The whole body laser scanning platform chosen for this research was Cyberware Laboratory Incorporated model WB4 triangulation laser scanner. This platform was chosen primarily due to the superiority of laser triangulation scanning for avatar purposes.

A complete body scan takes approximately 17 seconds, and captures both x-y-z coordinates and surface textures. Figure 5 shows the Cyberware WB4 body scanner.

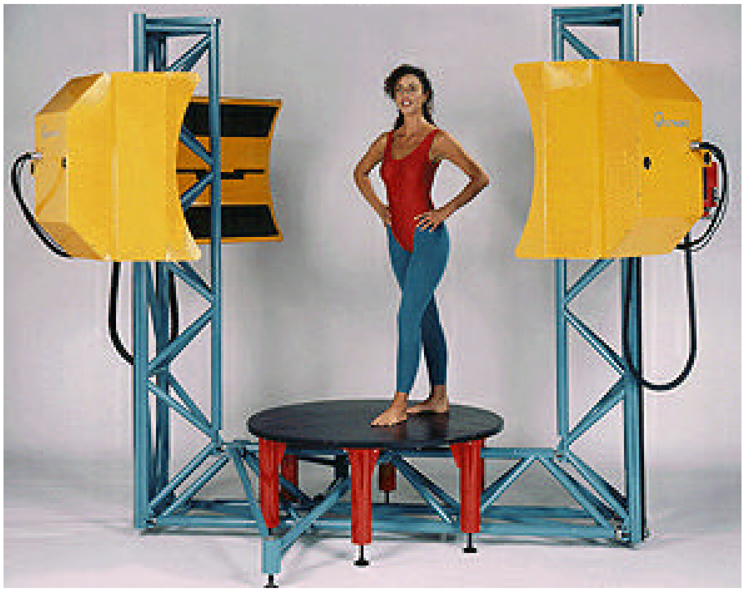


Figure 5. Cyberware Scanner Model WB4. From Ref. [5].

2. System Operation

Four yellow scanning heads are used to provide redundant data overlap to minimize the possibility of the subject inadvertently masking parts of their body from the lasers and thus preventing detection of those coordinates. These four scanning heads are mounted on vertical rails. A separate platform to support the subject allows for independent alignment of the heads. The scan begins with the heads at their topmost position. The heads travel down the rails, capturing body coordinate and Red-Green-Blue (RGB) texture information in one pass, which takes approximately 17 seconds. [Ref. 15]

The laser scan process is controlled via a computer graphics workstation. Scanner output is a “data cloud” of points, which form vertices for the rendering triangles. The resulting model is a single figure, without segmentation of any kind. It is in *.ply (Alias/Wavefront) format, which is in the public domain. This allows for easy file inspection, and for the possibility of constructing custom file translators. Cyberware has translators available that convert the scan data into the following file formats: 3D Studio, Digital Arts (SGI), DXF, IGES 128 NURBS, MOVIE.BYU (SGI), STL, SCR (SGI Mesh and Slice), ASCII, IGES (106, 110, 112, 124), Inventor, OBJ, Echo and VRML. [Ref. 16]

D. SUMMARY

This chapter discusses the methods, advantages and disadvantages of various technologies for 3D scanning of humans. Additionally, the whole body scanning system used for this research is examined in detail.

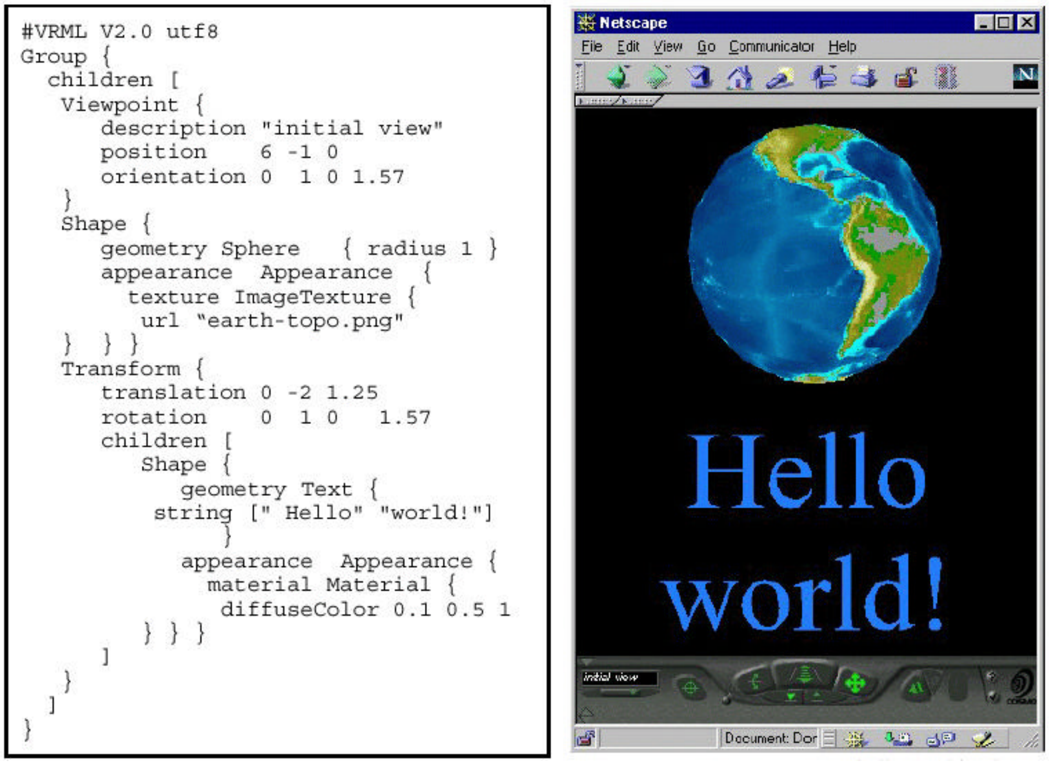
IV. RELATED WORK

A. INTRODUCTION

This chapter provides background on Virtual Reality Modeling Language (VRML), and on how VRML and Java work together. Further, it examines the Humanoid Animation 1.1 Specification (H-Anim 1.1 spec) and its canonical example, Nancy.wrl. Finally, various methods of human motion tracking are discussed, including a discussion of the inertial and magnetic limb segment trackers developed at the Naval Postgraduate School.

B. VIRTUAL REALITY MODELING LANGUAGE (VRML)

VRML provides a standard, platform-independent method of rendering 3D scenes across the Internet. It is a 3D scene description language for specifying virtual worlds. VRML supports both static and animated 3D/multimedia objects. VRML applications can imbed hyperlinks to many popular digital multimedia file types. The VRML specification is International Standards Organization (ISO) specification ISO/IEC 14772-1. Sample VRML output and source code is shown in Figure 6.



```
#VRML V2.0 utf8
Group {
  children [
   Viewpoint {
      description "initial view"
      position    6 -1 0
      orientation 0  1 0 1.57
   }
   Shape {
      geometry Sphere   { radius 1 }
      appearance  Appearance  {
        texture ImageTexture {
         url "earth-topo.png"
   }  } }
   Transform {
      translation 0 -2 1.25
      rotation    0  1 0   1.57
      children [
         Shape {
            geometry Text {
          string [" Hello" "world!"]
               }
            appearance  Appearance {
              material Material {
               diffuseColor 0.1 0.5 1
         } } }
      ]
   }
  ]
}
```

Figure 6. Example of Simple VRML Source Code and Output. From Ref. [17].

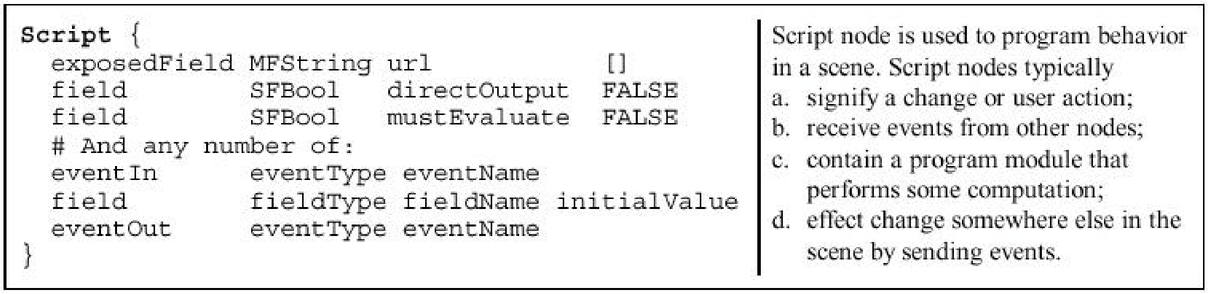
VRML scenes are constructed using nodes. These nodes are organized in a hierarchical fashion into a directed acyclic graph, or *scene graph*. VRML files end with *.wrl, or *.wrz if the file is gzip-compressed. The main method for the user to interact with a VRML world through a browser is via point and click. Thus VRML world content can contain embedded links just like traditional HyperText Markup Language (HTML). Typically VRML viewers, or browsers, are installed as plug-ins into popular 2D web browsers. [Ref. 17]

Four main components may be contained in a VRML file: the VRML header; Prototypes; Shapes (geometry and appearance), Interpolators, Sensors, Scripts; and Routes [Ref. 18]. Of these components, the only one required is the VRML header. Prototypes (PROTOs) are a powerful feature that provide for user-defined nodes, significantly increasing language extensibility. PROTOs may be combined into libraries, and are referenced using an external prototype (EXTERNPROTO) command, allowing for extensive code reuse. Shape nodes can contain both geometry and appearance nodes. Geometry nodes contain information on how the 3D object is constructed, and may be primitives such as a cylinder, cone, cube, or sphere, or may be text and indexed face sets. Appearance nodes describe how 3D objects look, and may include the object's color, texture, and transparency level. Interpolators allow for key frame animation. Sensors are the means by which the user interacts with the virtual world. Script nodes provide an interface for the VRML world to interact with a program script, such as Java or JavaScript. This ability for VRML to connect with powerful programming languages provides much flexibility, and is crucial for performing complex animations and network communication. Finally, routes define connections between nodes and fields, allowing for pre-defined events to be passed along the route to initiate program actions or animations.

C. VRML AND JAVA WORKING TOGETHER

By combining the authoring abilities of VRML with the programming resources of Java, a powerful hybrid is created that is more than the sum of its parts. Simple 3D content creation can be married with complex animated behaviors to give intricate results.

VRML and Java communicate via Script nodes, which contain Java functionality. Script nodes appear in the VRML file, and allow for connecting Java variables to VRML fields. Java classes must import vrml.* class libraries contained in the DIS-Java-VRML package in order to provide type conversions between Java and VRML. To interface properly with the VRML browser, Java classes used by Script nodes must extend the vrml.node.Script class. The basic Script Node interface is shown in Figure 7. [Ref. 17]



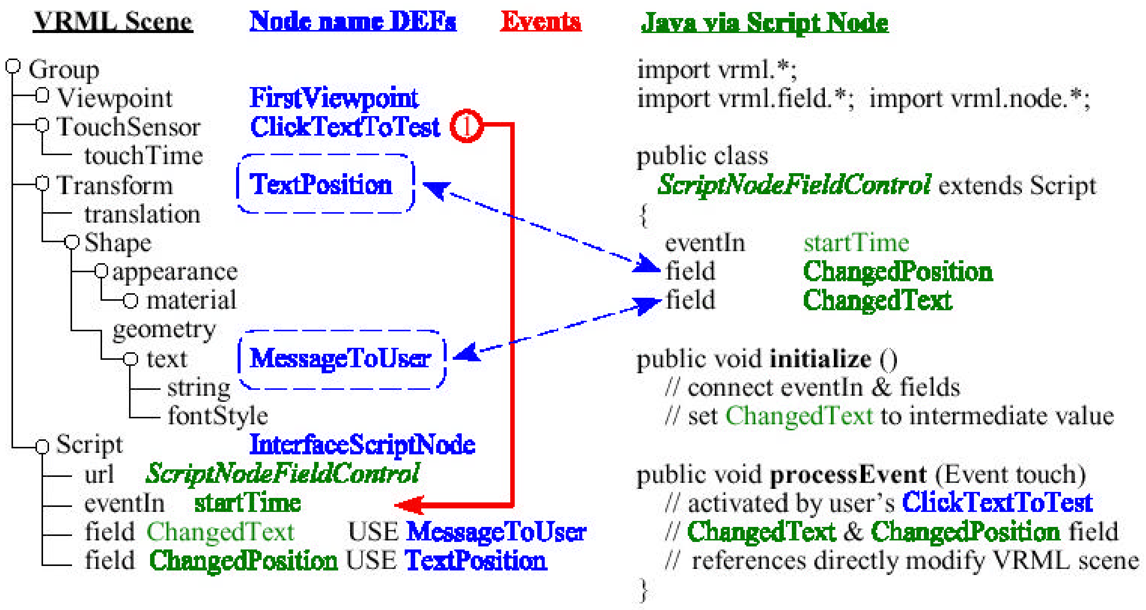
```
Script {
  exposedField MFString url            []
  field        SFBool   directOutput   FALSE
  field        SFBool   mustEvaluate   FALSE
  # And any number of:
  eventIn      eventType eventName
  field        fieldType fieldName initialValue
  eventOut     eventType eventName
}
```

Script node is used to program behavior in a scene. Script nodes typically
a. signify a change or user action;
b. receive events from other nodes;
c. contain a program module that performs some computation;
d. effect change somewhere else in the scene by sending events.

Figure 7. Script Node Interface. From Ref. [19].

The data type "exposedField" indicates that the associated variable has public access, whereas the "field" data type provides private access to the respective variable. The exposedField data member "url" contains the location of the java class file. This location may be locally on the hard drive or the Internet. For robustness, several urls may be entered so if the browser cannot find the named file in the first location, it will automatically look in the next location. The fields "directOutput" and "mustEvaluate" are hints to the browser on how to optimize performance. If directOutput is set to FALSE, the script only passes events and does not modify VRML nodes directly. Conversely, when directOutput is TRUE the script has permission to modify VRML nodes via the respective fields. If mustEvaluate is FALSE, the browser may postpone updating for rendering optimization. Setting this value to TRUE forces the browser to update when fields are modified. The data types "eventIn" and "eventOut" are events. Events are what provide VRML scenes their interactivity and fluidity. Events are time-stamped values of data types, and eventIn data types must match exactly the eventOut data types. When a pre-defined event is triggered, the value of the variable is sent (along with a time-stamp) from the eventOut connection to the associated eventIn connection.

An example is shown in Figure 8. Upon starting the VRML scene, the associated java class identified by the script node's "url" field is accessed, and its public method "initialize()" is called automatically. In this method, the fields passed by reference from the VRML file are connected to the "eventIn". The programmer may also perform any initialization that is deemed necessary, such as positioning or content changes. When the user activates the TouchSensor "ClickTextToTest" by clicking on the text with the mouse, an event and time-stamp is sent from touchTime's eventOut to the script node's eventIn "startTime". This calls the script node's pubic method "processEvent". The programmer can then perform any java functionality that is desired, and modify the fields passed in by reference accordingly. In this example, both the content and position of the text string is modified. [Ref. 17]



```
import vrml.*;
import vrml.field.*;  import vrml.node.*;

public class
  ScriptNodeFieldControl extends Script
{
   eventIn       startTime
   field         ChangedPosition
   field         ChangedText

public void initialize ()
   // connect eventIn & fields
   // set ChangedText to intermediate value

public void processEvent (Event touch)
   // activated by user's ClickTextToTest
   // ChangedText & ChangedPosition field
   // references directly modify VRML scene
}
```

Figure 8. Example Script Node and Java Interaction. From Ref. [17].

D. THE HUMAN ANIMATION 1.1 SPECIFICATION

The Humanoid Animation Working Group of the Web3D Consortium developed the H-Animation 1.1 spec. The working group states the following charter [Ref. 20]:

> Our aim is to specify a way of defining interchangeable *humanoids* and *animations* in standard VRML 2.0 without extensions. Animations include limb movements, facial expressions and lip synchronization with sound. Our goal is to allow people to author humanoids and animations independently…

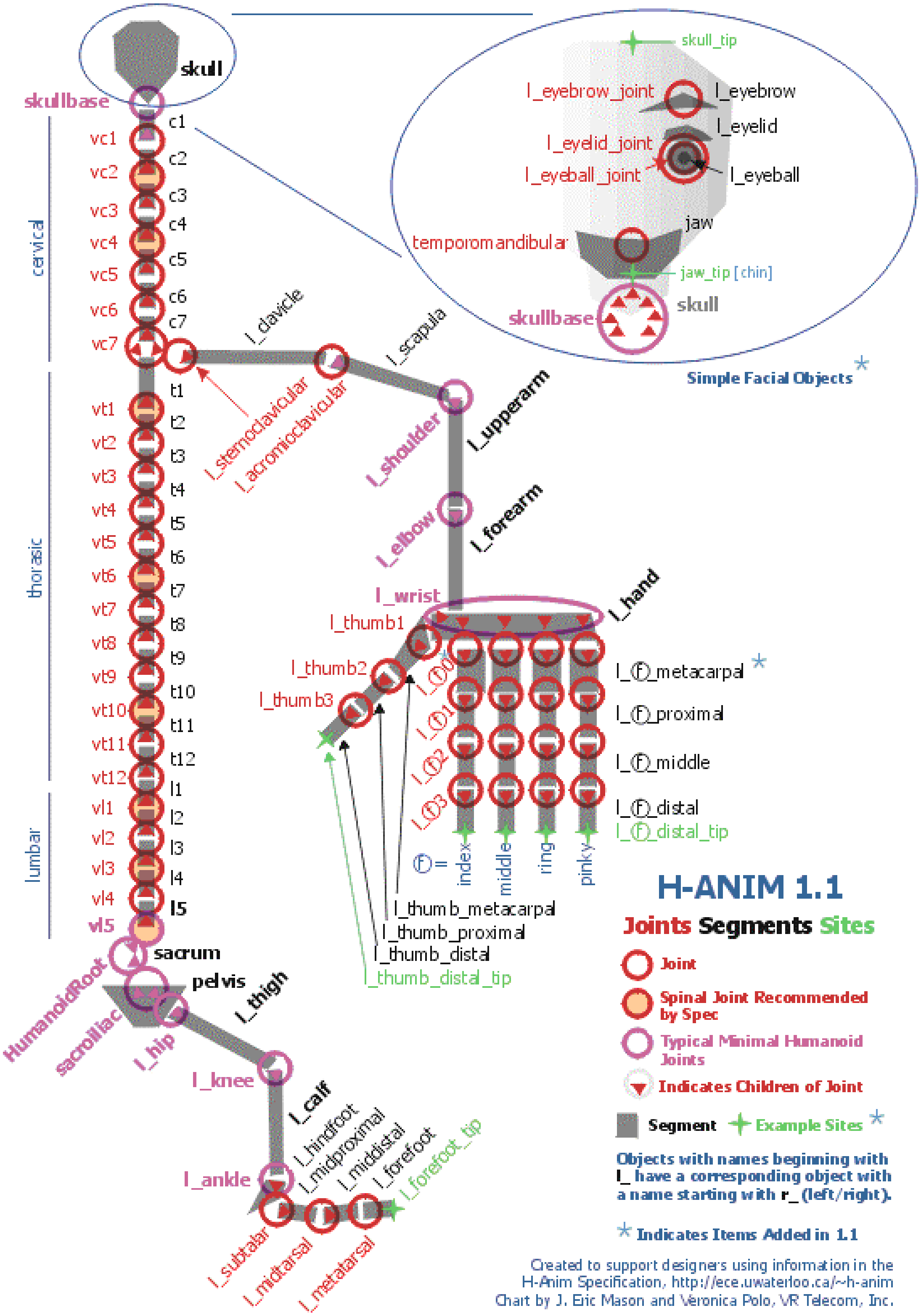
Although originally restricted to VRML 2.0, the working group's goal has grown to providing virtual humanoid form and behavior regardless of the authoring tool used, and allowing for the interchangeability of virtual humanoids. No assumptions were made concerning the application that would use the humanoids. One example of the specification's flexibility is its appearance in High Level Architecture (HLA), where it has been developed as a Federation Object Model (FOM) [Ref. 21].

The H-Anim 1.1 spec has as its root a single Humanoid node. This node serves the following purposes:

- Stores human-readable data about the humanoid such as author and copyright information.
- Provides a top-level Transform field for positioning the humanoid in the environment.
- Stores references to all the Joint, Segment and Site nodes.
- Serves as a "wrapper" for the humanoid.

Joint nodes are arranged in a strictly defined hierarchy. They may contain other joint nodes, or segment nodes. Segment nodes describe the portion of the body connected to the associated joint, and may contain Site nodes and Displacer nodes. Site nodes contain location information relative to the segment, and can be used for placing clothing, jewelry, or other items on the segment. Site nodes may also be used as a "manipulator handle" for inverse kinematics applications. Displacer nodes are simply grouping nodes, allowing the programmer to identify a collection of vertices as belonging to a functional group for ease of manipulation. [Ref. 20]

The H-Anim 1.1 Spec defines the "at rest" position, which specifies all joint rotations to be zero. Additionally, it specifies that the origin be located between the feet of the humanoid at ground level, and that the humanoid face the +z direction, with +y being up and +x to the left of the humanoid. Just as important, the specification provides naming conventions for 94 joints and their associated segments, allowing for an extremely complex avatar (see Figure 9).



Created to support designers using information in the H-Anim Specification, http://ece.uwaterloo.ca/~h-anim
Chart by J. Eric Mason and Veronica Polo, VR Telecom, Inc.

Figure 9. H-Anim Spec 1.1 Hierarchy. From Ref. [20].

E. NANCY: THE H-ANIM 1.1 STANDARD

Nancy.wrl was chosen as the foundation on which to build the avatar used in this research, and is the canonical example of H-Anim 1.1. The author of *Nancy* is Cindy Ballreich, who grants permission for its non-commercial usage with proper credit and use of the 3Name3D name and logo. In the case of this thesis, Nancy was fundamentally modified such that maintenance of the 3Name3D name and logo would have proven challenging. Cindy Ballreich kindly granted permission for the use of Nancy for this research without the company name and logo.

Nancy contains 17 joints, 15 segments and four default viewpoints. Four pre-scripted behaviors are included: stand, walk, run and jump. The user clicking on the appropriate text inside of the VRML world activates these behaviors. See Figures 10-13.

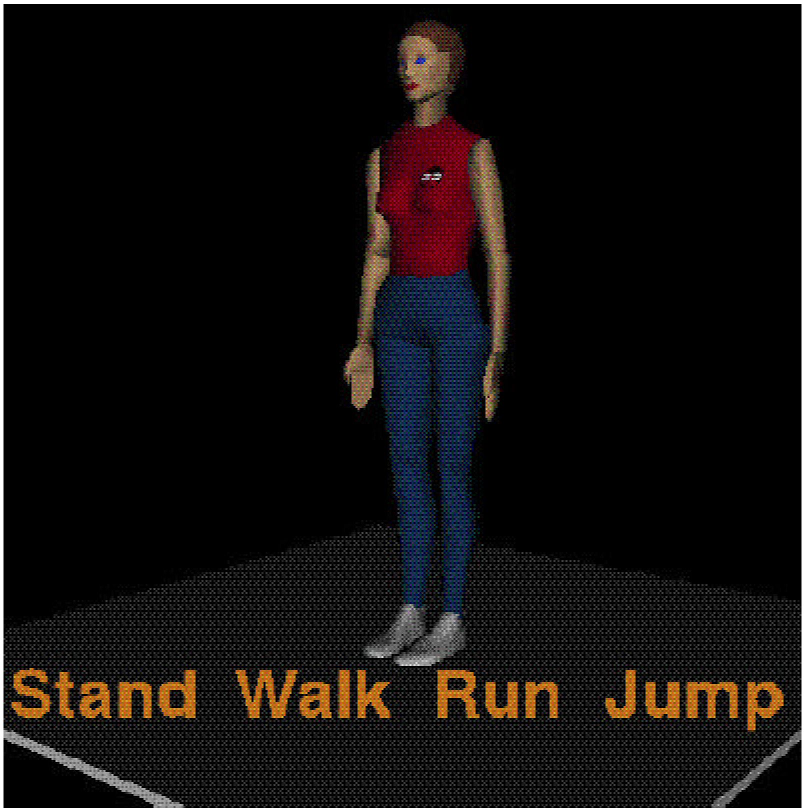


Figure 10. Nancy Demonstrating the Stand Behavior.

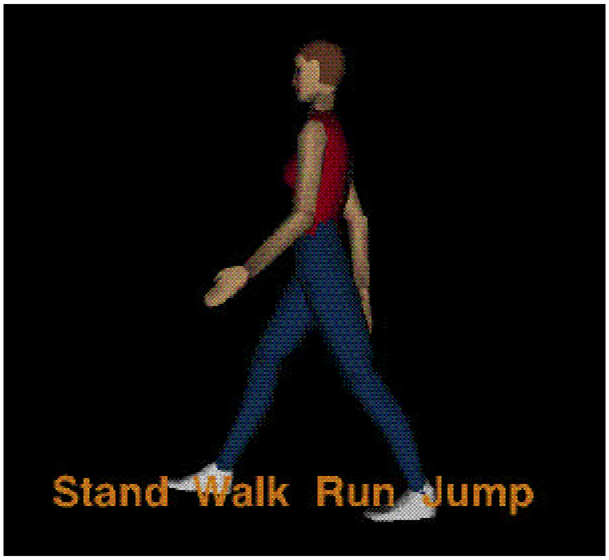


Figure 11. Nancy Demonstrating the Walk Behavior.

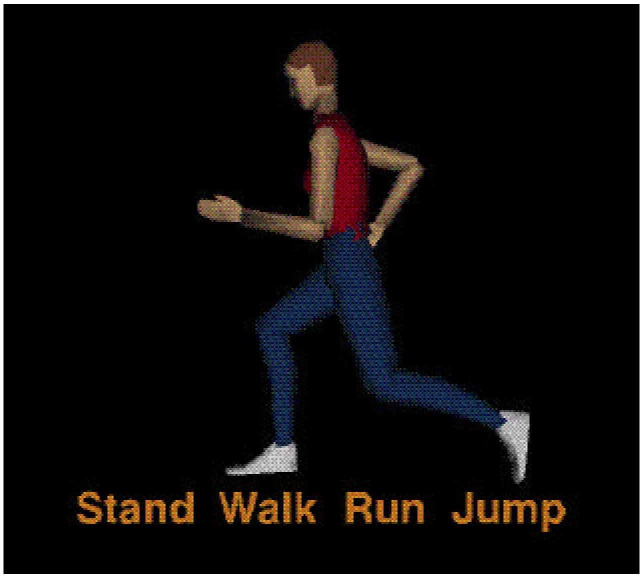


Figure 12. Nancy Demonstrating the Run Behavior.

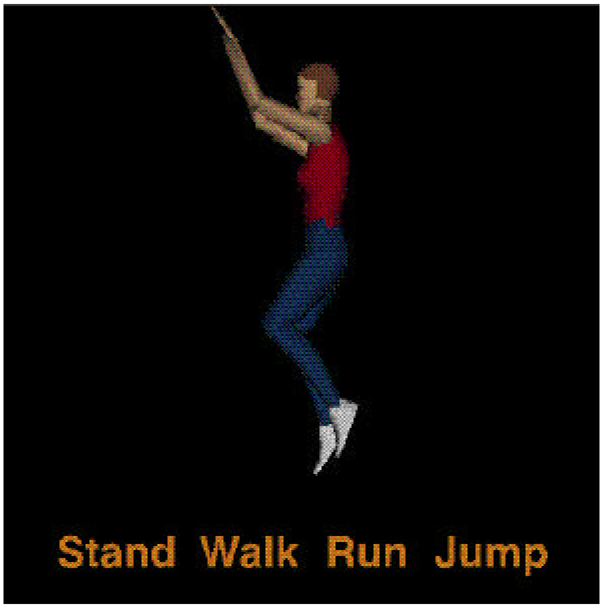


Figure 13. Nancy Demonstrating the Jump Behavior.

F. HUMAN MOTION TRACKING TECHNOLOGIES

This section provides an overview of some of the human motion tracking methods available at the time of this research. The objective is not to undertake an exhaustive study of the field of motion tracking, but to discuss some of the more prevalent technologies and their respective limitations.

1. Mechanical Trackers

Mechanical tracking is capable of not only tracking the movement of the user, but also of permitting the virtual environment to make itself felt through the use of haptic feedback. Since mechanical tracking is relatively accurate, much research has been done in using mechanical tracking as a calibration standard for various other tracking systems [Ref. 22]. Mechanical tracking can usually be categorized into of one of two forms: body-based and ground-based. [Ref. 7]

Body-based mechanical tracking is performed by having the user wear a mechanical frame, or exoskeleton (see Figure 14). Angle measuring devices, called goniometers, are located at exoskeleton joint locations. By measuring the joint angles of the exoskeleton, user limb orientation is obtained.

One disadvantage of mechanical tracking is that since only the mechanical frame is tracked, errors are introduced if the exoskeleton shifts position on the body. Also, goniometer alignment with the joints is difficult. The goniometers are located externally to the joint, and are therefore displaced from the joint by some offset amount. This offset must be taken into account since it can introduce orientation errors. [Ref. 7]

Another disadvantage of mechanical tracking is encumbrance. Not only must the user bear the weight of the exoskeleton, but it may also be impossible to obtain certain positions due to the size or shape of the device. These difficulties tend to detract from the user's sense of presence, and severely limit the scope of the user's interaction with the virtual world. Although accurate and relatively inexpensive, mechanical tracking of several users in a single, shared volume is problematic due to both interference of the mechanical linkages and limited range. [Ref. 7]

2. Magnetic Trackers

For real-time applications, magnetic tracking is the most prevalent. Possessing reasonable accuracy with little or no obstruction problems, these relatively inexpensive systems track both segment position and orientation with body-mounted sensors that measure a spatially varying magnetic field.

Since the systems are magnetic, they possess disadvantages common to any magnetic-field device. As sensor distance from the source increases, magnetic field strength decreases in power inversely with the square of the distance. This effectively limits the useful range of magnetic tracking, usually to less than ten feet. Additionally, orientation and position errors due to distortions of the spatial magnetic field increase with the fourth power as the source distance increases. This results in a non-constant error that varies according to sensor position and orientation relative to the source. Further, nearby metal objects can interfere, causing permutations and even obstructions of the magnetic field. Another disadvantage to magnetic tracking is latency. Vendor latency data varies enormously, depending on the application. Finally, electrical components generate their own magnetic field, which may induce noise and erratic magnetic field behavior. [Ref. 22]

3. Optical Trackers

Optical tracking is quickly catching up to magnetic tracking in terms of popularity. Currently, the main application of optical tracking is animation requiring extensive off-line processing. It has been used in few real-time applications. The film industry relies on this technology almost exclusively, as it is highly reliable under controlled conditions.

Since optical tracking depends on various light sources, the systems are highly susceptible to interference from other light sources near the same frequency. Also, since detection of optical sensors requires line of sight (LOS) the systems are vulnerable to occlusion, making tracking of multiple people in a common work volume difficult. Further, some types of light severely limit the range of some optical tracking systems.

Optical tracking systems fall into one of three categories. *Image-based systems* track position and movement by using multiple video cameras to track pre-selected sensors attached to the user. *Pattern-recognition systems* sense the distortion of a projected pattern of light to track position and orientation. This is a motion analog to Moire scanning which was discussed earlier in this thesis. *Structured light and laser systems* have been promising, but thus far have not enjoyed the attention of researchers to the same extent as the other optical tracking systems. [Ref. 7]

4. Acoustic Trackers

Acoustic, or ultrasonic trackers provide reasonable update rates and accuracies, and are less expensive than magnetic trackers. However, just as magnetic trackers were limited by the underlying physics of magnetism, acoustic trackers are limited by the physics of sound. Although ultrasonic systems have longer ranges than magnetic systems, they must maintain line of sight making obstruction and shadowing a problem. The range of the system is dependent on wavelength. If wavelength is too short, acoustic interference is minimized but range is minimized as well. On the other hand, if wavelength is too long, latency becomes unacceptable and distance resolution suffers. The middle frequency band that remains is susceptible to acoustic interference from metallic objects, in addition to the echoes and reflections to which all sound is vulnerable. [Ref. 24]

5. Inertial and Magnetic Tracking

Inertial and magnetic tracking is one of the newer motion tracking technologies. Although it has been used for tracking user head positions in various virtual reality applications, until recently it had not been used for full body tracking. Professor Eric Bachmann at the Naval Postgraduate School developed one of the first platforms for using Magnetic Angular Rate Gravity (MARG) tracking as a full body motion tracking system. [Ref. 7]

With recent advances in micro-machined and miniaturized technology, inertial tracking has become an affordable and accurate option. Unlike the other sensing technologies discussed, inertial trackers contain no inherent latency and therefore should be more accurate than their counterparts.

With inertial tracking, angular rate data is integrated to determine segment orientation. If this data were used alone, error would be introduced over time as bias and drift errors accumulated. However with the addition of accelerometers to sense the gravity vector and magnetometers to sense the local magnetic field, the inertial signal can be corrected and the errors minimized. The MARG sensors developed by Bachmann contain a separate accelerometer, rate sensor and magnetometer for each coordinate axis.

One drawback to the system developed by Bachmann is that only orientation is tracked, not position. Mobile platforms, such as submarines, integrate accelerometer data to obtain position, but their accelerometers are much larger and significantly more expensive. Currently, such techniques may only be applied for short time periods with the small, low-grade sensors used for MARG tracking before drift introduces significant error. [Ref. 23][Ref. 24]

G. SUMMARY

This chapter discussed the Virtual Reality Modeling Language (VRML), and examined the powerful combination of VRML and Java working together. The Humanoid Animation 1.1 Specification and its canonical exemplar, Nancy.wrl were also discussed. Finally, a brief overview of current human body motion tracking was provided, including the method chosen for this research, inertial and magnetic (MARG) tracking developed at the Naval Postgraduate School.

V. INITIAL DEVELOPMENT EFFORTS

A. INTRODUCTION

This chapter discusses the challenges involved in reducing the complexity of laser scan output, translating between file formats and partitioning the resulting data cloud into body segments. Constructing an avatar from these body segments is then examined. Finally, the process of making Java and VRML work together, and incorporating networked, real-time control is discussed.

B. HANDLING LASER SCAN DATA

1. Reducing Laser Scan Output Complexity

The first challenge of this research was to simplify the laser scan data set. The raw output data consists of approximately 150,000 polygons. When translated into ASCII text, the size of the file is over 50 megabytes. This large data set is extremely unwieldy. If it could be rendered at all, use of a model of this nature in 3D worlds under current technology would be extremely inefficient, resulting in very slow frame rates. Figure 15 shows the original, unreduced laser scan output.

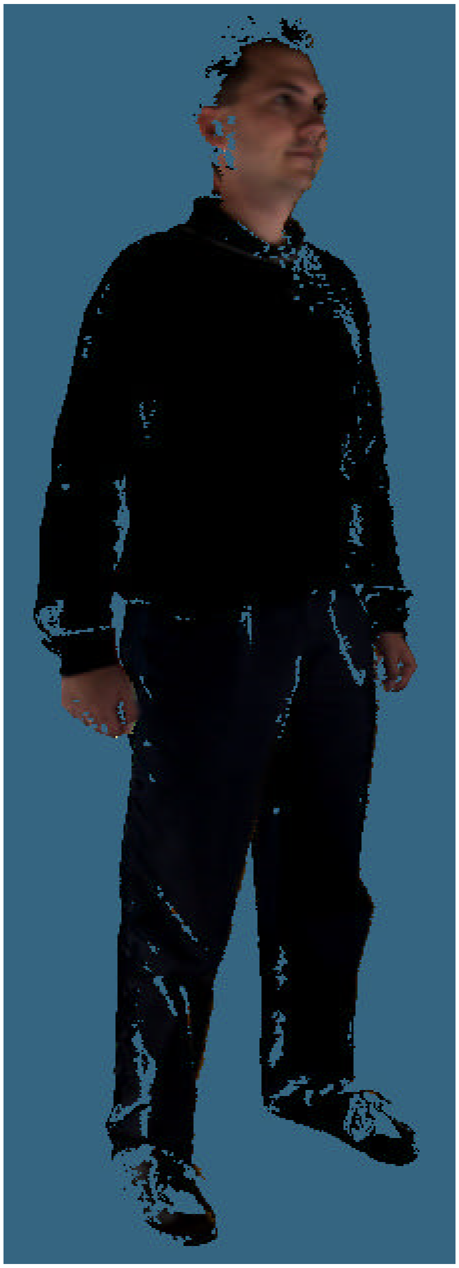


Figure 14. Initial Laser Scan Output of the Author (150,000 Polygons).

For realistic rendering of humans in most applications, only 4000 to 5000 polygons are necessary. Texture mapping further assists in lowering the required polygon count, as the overlying texture adds greater surface detail.

Laser scan output is in *.ply format, an Alias/Wavefront file type. Since this format is open source, it is possible to write custom polygon reduction algorithms. As the scope of this thesis is avatar construction and real-time avatar control, it was considered beyond the scope of this research to create a custom algorithm for polygon reduction. Instead, a proprietary software package from Cyberware Laboratories called "Decimate" was used. The Decimate software can reduce model polygon count from the initial number of 150,000 to whatever the user specifies [Ref. 26]. For this research, a target polygon count of 10,000 polygons was used. Polygon reduction from 150,000 polygons to 10,000 polygons was completed in less than 15 seconds on a Pentium III, 550-megahertz system. The resulting ASCII text file size was approximately 750 kilobytes.

2. Translating Between Files

The *.ply file format is intended for animation software packages. For real-time rendering into 3D worlds a different file format is needed. Specifically, the Virtual Reality Modeling Language (VRML) format (*.wrl) was chosen for reasons discussed in Chapter III.

As was the case for polygon reduction, custom translators could be written since *.ply is open source. Again, file translation was considered beyond the scope of this research. Additionally, the Decimate software used for polygon reduction includes several file translators, including VRML. The disadvantage in using the VRML translator that comes with Decimate was that original texture information is lost when converting from *.ply to *.wrl format, thus, if texture mapping is desired it must be supplied by the modeler. Figure 16 shows the reduced, VRML model obtained from translation.

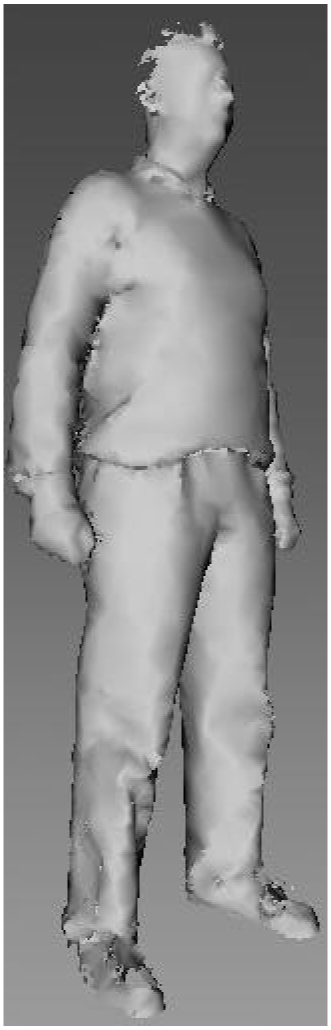


Figure 15. Translated VRML Avatar (10,000 Polygons).

Two things should be noted about Figure 16. The first is the lack of texture information, for the reason discussed in the preceding paragraph. The second item to note is that the figure is one complete piece. That is, the figure is not articulated or segmented in any way. The challenge of partitioning the avatar into appropriate body segments is discussed next.

3. Segmenting The Avatar

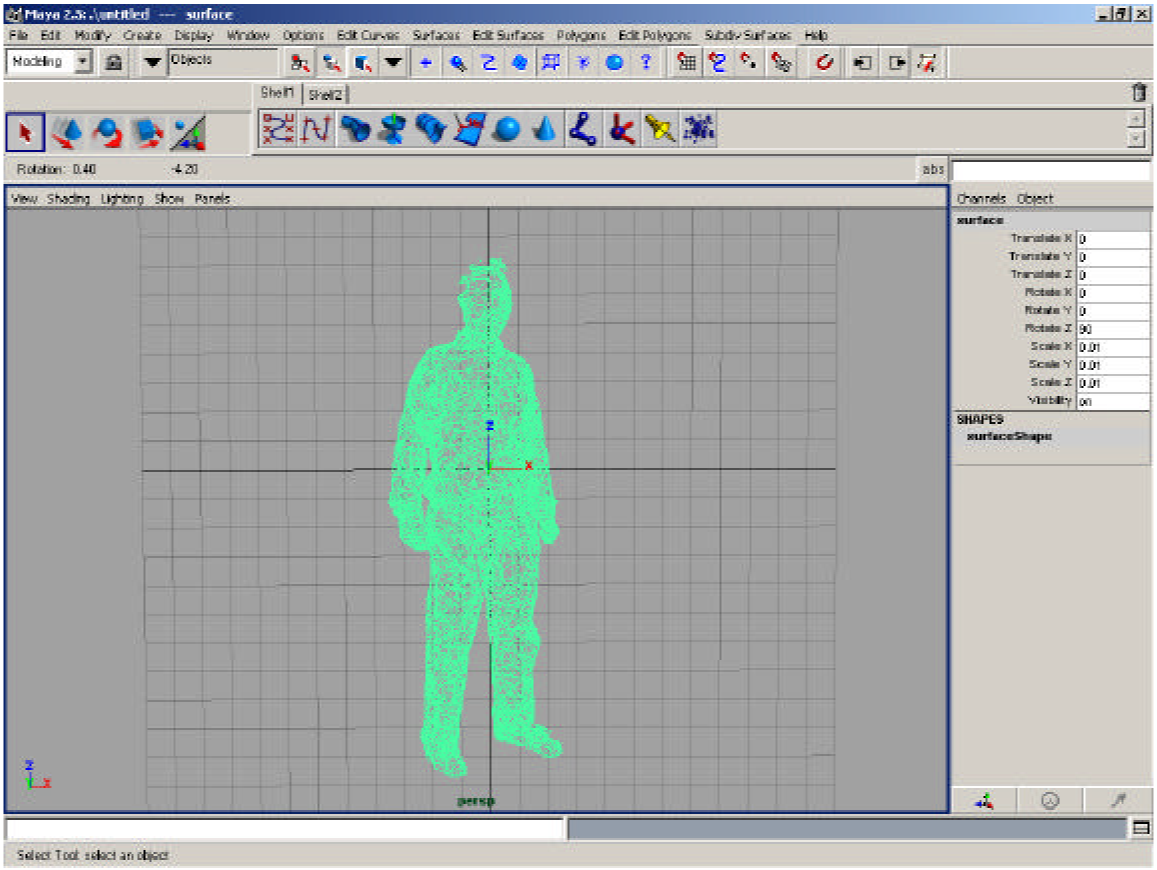
For a human model to be able to mimic the full range of motion of a human, it must be segmented in the appropriate places. Three approaches were considered for avatar segmentation.

The first approach that was considered is completely automated. If it is possible to infer the location of important body joints from the model data, then code may be written to extract such information and apportion body segments appropriately. One possibility is that when the individual gets scanned, they stand in such a way so that important joints are bent past some critical angle. The segmentation code could then look for sufficient change in the direction of surface normals, thereby indicating a bent joint. Since the initial laser scan was performed at the beginning of this research there were no preferences for scan position, and thus all joints are straight in the model. This being the case, completely automatic joint selection was not feasible.

Partial automation was considered next. If average body segment lengths were known, it would be possible to insert joints automatically at the average locations. The drawback to this method is accuracy. Joint locations can vary widely from subject to subject, so models using this method would be susceptible to inaccurate segmentation, resulting in unbelievable avatars. Since one of the major objectives of this research is to provide realistic, believable avatars this method of partial automation was discarded.

The final method of avatar segmentation that was considered, and ultimately used in this research, was completely manual. An operator imports the laser scan data into 3D-rendering capable software, manually selects the segments, and then exports the virtual body parts for use as avatar building blocks. A disadvantage to this method is that it is time consuming. Several hours are required for an operator to segment a model. Another disadvantage is that model segmentation is somewhat arbitrary. One operator may segment a model very differently than another operator, especially if joint position is unclear, as in the initial scan. Lastly, this method requires operators trained in the use of whichever software package is selected.

The software package used for avatar segmentation in this research was Maya, from Alias/Wavefront. Maya has been used extensively in the film industry to provide lifelike animation, and is adept at handling 3D objects [Ref. 27]. Maya can import and export *.obj files, allowing for segmentation processing. A file translator provided with Cyberware's Decimate was used to convert the polygon-reduced laser scan from *.ply to *.obj format. The *.obj file was then imported into Maya, and 3D selection of body segments was performed (Figure 17). After each body segment was selected, it was exported as a separate *.obj file. Unfortunately, at the time of this research Decimate did not contain translators to convert directly from *.obj to *.wrl (VRML), so it was necessary to first convert each body segment from *.obj to *.ply, then *.ply to *.wrl.



Maya 2.5: .\untitled --- surface
File Edit Modify Create Display Window Options Edit Curves Surfaces Edit Surfaces Polygons Edit Polygons Subdiv Surfaces Help
Modeling
Objects
Shelf1 Shelf2
abs
View Shading Lighting Show Panels
Channels Object
z
x
persp

Rotation: 0.40 -4.20

surface

Translate X	0
Translate Y	0
Translate Z	0
Rotate X	0
Rotate Y	0
Rotate Z	90
Scale X	0.01
Scale Y	0.01
Scale Z	0.01
Visibility	on

SHAPES
surfaceShape

Select Tool: select an object

Figure 16. Reduced Laser Scan Imported Into Maya.

C. CONSTRUCTING THE AVATAR

After segmentation in Maya, the model consisted of several VRML files, one file for each body segment. Each file contains VRML Shape nodes, with geometry data called an “IndexedFaceSet.” This geometry data contained x-y-z coordinates for each point to be rendered, along with indexing information indicating the order in which to render the points. Order is important because VRML supports back-face culling for efficiency. Back-face culling is an operation performed by rendering engines that only draws the external faces of objects. Since observer viewpoint is seldom concerned with internal views of a "solid" object, drawing only the external sides can significantly reduce the processing workload, and result in higher frame rates. VRML, like most current 3D rendering engines, determines which sides are external and which are internal by the order in which points are drawn.

The Humanoid Animation Specification 1.1 (H-Anim 1.1) canonical example Nancy.wrl, created by Cindy Ballreich, was used (see Figures 10-13). Nancy's segments were constructed using indexed face sets and indexing data. To construct the laser-scanned avatar, the information (x-y-z coordinates and indexing data) from each of Nancy's original segments was replaced with the corresponding information from each of the segment VRML files exported from Maya. Each of the new segments was scaled, and connected together by appropriate rotation and translation. The result is an articulated VRML/H-Anim 1.1 avatar originating from a laser scan, capable of scripted behaviors. As discussed earlier, texture information is not present due to inadequacies with Cyberware Laboratory's *.ply to *.wrl translator, so Nancy's default colors were used initially as shown in Figure 18.

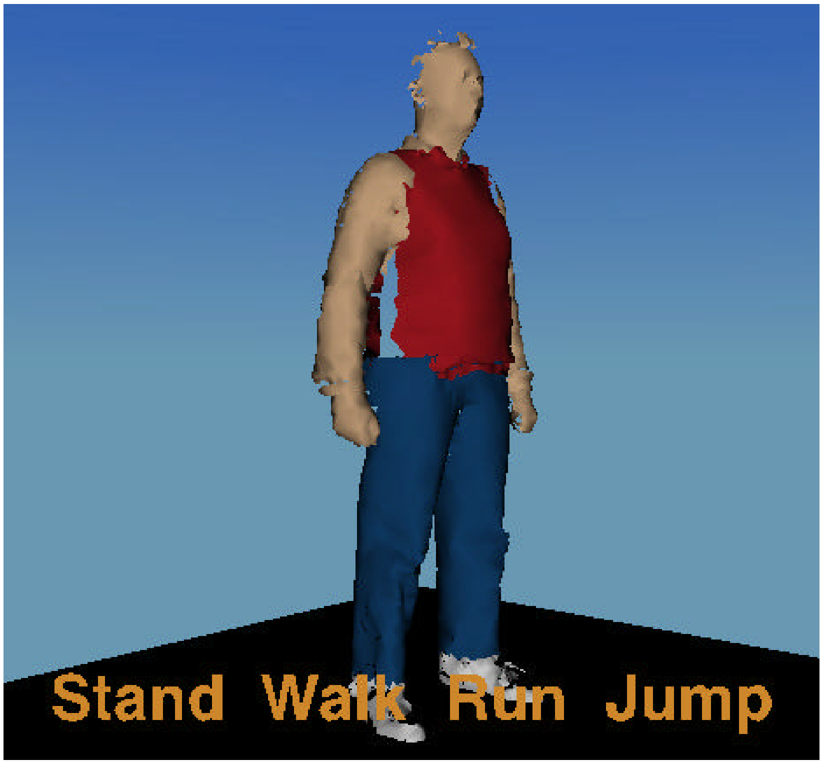


Figure 17. Initial VRML/H-Anim 1.1 Avatar From Laser-Scan (Untextured).

Two things should be noted about Figure 18. The first is that there are no polygons above the hairline. The physical characteristics of hair yield a poor return signal during the laser-scan process, resulting in the loss of coordinate information for the portions of the skull covered by hair. The second thing to note is the gap appearing between the right arm and the torso. Due to the posture of the human subject during the scanning process, the arms effectively blocked the laser signal from reaching the left and right sides of the torso, resulting in loss of coordinate information. Similar situations exist with other body segments, such as the legs.

Both the hair-interference and segment-shadowing problems may be compensated for using standard 3D editing techniques, either using popular animation programs such as Maya or directly by point editing in VRML. The segment-shadowing problem may also be minimized by placing the human model in a posture that maximizes exposed surface area before the laser scan begins.

A third problem with the avatar, which is not immediately apparent from Figure 18, is one of joint visual connectivity. Since the avatar was obtained from a static laser scan, when segments are moved from the initial scan position "tears" can be seen at the joints where segments are connected. For example, consider a human arm. When someone moves a forearm in the physical world, the skin, muscles and tendons stretch to accommodate the varying positions of the forearm relative to the upper arm. In the final avatar created by this research, when segments move relative to each other the surface topology does not stretch or otherwise compensate for varying segment positions, resulting in visual tearing between segments during some avatar movements.

For added realism, a standard green camouflage pattern was applied to each avatar segment, with the exception of the head, hands and feet. For the head, a different process was used.

3Q, incorporated [Ref. 28] specializes in optical triangulation scanning of the human face. They have booths in some software entertainment stores that perform digitization of faces. The digitized face can then be imported into a variety of popular computer games, and also contains a VRML rendition. This VRML output was used to replace the existing avatar skull obtained from the laser-triangulation scan performed by Cyberware Laboratories.

The end product is a fully articulated, texture-mapped avatar that is capable of scripted movement via an international standard, H-Anim 1.1. See Figure 19.

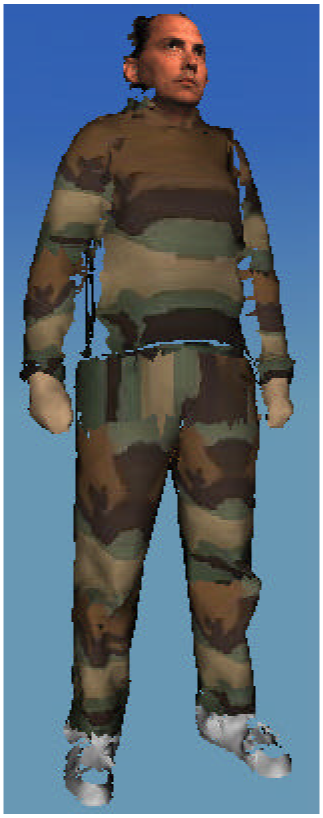


Figure 18. Texture Mapped, Articulated Avatar Capable of H-Anim 1.1 Scripted Movement.

D. USING JAVA TO PROVIDE REAL-TIME NETWORKED CONTROL

Although capable of scripted movement, the final product must also be controllable via network updates. Adding the open-source DIS-Java-VRML [Ref. 29] package to Java makes communication between VRML and Java possible. In this case, VRML renders the 3D scene and Java handles the networking. Refer to chapter 3 for an in-depth discussion on how VRML and Java can work together.

The network protocol chosen for this implementation is the User Datagram Protocol (UDP). UDP is connection-less. Although not as reliable it is faster than

Transmission Control Protocol/Internet Protocol (TCP/IP) [Ref. 30]. Strict packet accountability was deemed unnecessary for this research, since segment orientations are typically updated at approximately 100 hertz [Ref. 7].

Two UDP approaches were considered. The first approach examined was the built-in UDP functionality provided by DIS-Java-VRML, which provides an easy-to-use UDP class called the Protocol Data Unit (PDU). Twenty-seven PDU fields are defined in the 1995 IEEE Standard for DIS-Application Protocols [Ref. 31]. Since the only information needed to be passed is a field containing segment identification, and four other fields containing orientation information the PDU was dismissed as being too heavyweight and therefore inefficient for this application. The second approach involved custom packet design. Although this technique involved more coding and design, it was ultimately selected due to its superior efficiency, since a packet could consist of only five fields versus the twenty-seven fields contained in the PDU.

The final code consists of the following files: JavaDutton.wrl (the file containing the VRML content), greenCamo.jpg (contains the green camouflage texture map), clone.gif (the face picture to be texture mapped onto the avatar's face), ScriptNodeFieldControl.java (VRML/Java interface), ClientProgram.java (receives UDP segment orientation updates) and QuatToEuler.java (converts the quaternions received from the MARG sensors to Euler angles for VRML use).

An additional class, ServerProgram.java, was written to support testing. Unfortunately, at the time of this writing the body tracking software developed at the Naval Postgraduate School only outputs to text files [Ref. 7]. In anticipation of network updates, the ServerProgram class simulates direct networking by parsing a pre-recorded body tracking session, wrapping the data into a UDP packet, and sending it over the network. When the ClientProgram class receives data over the network, it is unaware that the source was originally a text file. One drawback to this method is speed: parsing the data from a text file slows down the update process considerably, resulting in an animation playback that is an order of magnitude slower than animation being driven by pure network updates.

To run the networked VRML avatar, all of the java class files must be in the same directory as JavaDutton.wrl, greenCamo.jpg and clone.gif. Assuming a VRML viewer plug-in has been installed in the web browser, the user double-clicks on JavaDutton.wrl. The initial 3D scene is rendered, and ScriptNodeFieldControl is automatically called. ScriptNodeFieldControl accepts and initializes the avatar segment nodes along with a text message that is rendered in front of the avatar. ClientProgram is automatically called by ScriptNodeFieldControl as a separate thread, which then listens for UDP packet updates. When users are ready to receive network updates, they click on the text in front of the avatar in the VRML scene. The text message changes, and body segment orientations are continuously updated from an array containing the most recent orientation data. For testing, ServerProgram was started, with a command line argument containing the filename of the pre-recorded body tracking data. ServerProgram parses the input file, calls QuatToEuler to convert quaternions to equivalent Euler angles, and sends the data over the network in the form of UDP packets. When ClientProgram receives a UDP packet, it unwraps the packet and calls an update method in ScriptNodeFieldControl, which then updates the array containing the most recent segment orientations. The next time the VRML scene graphics are refreshed, these most recent orientations are read from the array, thus updating the avatar's motion.

E. SUMMARY

This chapter examined the process used to reduce the polygon count of the initial laser scan, translate the data into various file formats, and partition the data into body segments. Additionally, avatar construction from these body segments was discussed. Finally, the process of making Java and VRML communicate with each other, and providing for networked, real-time control was examined.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS AND RECOMMENDATIONS

A. GENERAL THESIS CONCLUSIONS

Construction of an articulated avatar from laser scans for use in 3D networked virtual environments has been achieved. The resulting avatar resembles the original human to a scale of millimeters and runs efficiently on current standard computer desktop systems. The 3D engine can be user and programmer-friendly, platform independent and open-source. Avatars can be driven by either scripted behaviors, or by real-time control via networks.

B. SPECIFIC CONCLUSIONS AND RESULTS

1. Constructing Anatomically Accurate Avatars

The H-Anim 1.1 exemplar Nancy.wrl, created by Cindy Ballreich, was used as the foundation and served as inspiration for this project. By replacing Nancy's coordinate and indexing data with laser scan data, exact anatomical avatars can be constructed that conform to an international human animation specification.

2. Flexible Avatar Control

Avatar control is possible by either programmed or real-time input. Programmed, or scripted, input follows the H-Anim 1.1 specification. Real-time input may be accomplished over a network, with control devices sending UDP packets containing limb segment orientation updates. Specifically, real-time networked control via wireless motion tracking sensors developed at the Naval Postgraduate School was implemented as proof-of-concept.

3. Source Code is Platform-Independent and Open-Source

By restricting computer source code to VRML and Java, the final product produced by this research will run on various platforms and operating systems via popular web browsers. All source code is open-source, allowing for both inspection and enhancement as technology progresses.

4. Simple 3D Authoring is Combined With Powerful Programming Capability

VRML is a high-level, easily understood 3D authoring language. Java is a powerful and widely used programming language. Highly complex results may be obtained when the two are combined, producing a product that is more than the sum of its parts. With gains in computing technology, both Java and VRML are approaching run-time speeds previously enjoyed only by low-level programming languages, allowing easy creation of intricate scenes and behaviors through relatively simple interfaces.

C. LESSONS LEARNED

The laser scan of the author was performed at the beginning of this research. During this thesis it became clear that the scan pose was less than optimum. Not only were body segments masking other portions of the body from the laser signal, but also all of the limb segments were straight, making it difficult to determine exact joint position during the segmentation process. With the knowledge gained during this research, it is recommended that future avatar scans be performed differently. First, limbs should be positioned in such a way as to minimize masking other portions of the body from the laser signal. Second, all of the major joints should be bent as close to 90 degrees as possible. Not only does this provide clear indication of joint location for manual segmentation, but it also provides a clear division between limb segments for possible automated segmentation.

D. RECOMMENDATIONS FOR FUTURE WORK

1. Automating Avatar Segmentation and Construction

Segmenting the avatar into appropriate body segments for articulation, and then constructing the avatar from these segments was by far the most time-intensive component of this research. Advanced knowledge of third-party animation software, in this case Maya, was required. Since segmentation was done manually, accurate and consistent joint separations were both difficult and time-consuming. If the initial scan pose is modified as discussed earlier, it should be possible to automatically determine joint location, based either on the relatively rapid change in surface normal direction or some other algorithm. These automatically generated segments could then be placed together using an avatar template. In this manner, articulated avatars could be constructed in a few minutes instead of a few days, allowing for rapid content creation.

2. Updating File Translators to Retain Texture Information

During the whole body laser scan, texture information as well as x-y-z coordinate information is obtained. Unfortunately, as discussed in Chapter V, the current file translators provided by Cyberware lose texture information when converting scan data to VRML. Cyberware is aware of this problem, and may resolve this issue in a later software release. Alternatively, since laser scan output is in open-source *.ply format, custom translators that properly retain texture information could be written. One possible approach would be to include custom file translation that retains texture information with the automatic avatar segmentation and construction process discussed earlier.

3. Keep Joints Connected in All Positions

Since the avatar was created from a static model, visual tearing occurs when the relative positions between limb segments are changed. This detracts from avatar realism and overall appearance. An important improvement would be to use displacers or meshes to keep limb segments cohesive through all ranges of motion. Some advanced features of VRML support such techniques [Ref. 18].

4. Increase Behavior and Motion Libraries

Since the capability of scripted avatar control is provided via H-Anim 1.1, an extensive library of ready-to-use behaviors would be of great benefit to virtual environment designers. Depending on the target application, pre-existing complicated behaviors could be imported with little or no significant extra development time, allowing for more rapid and believable content creation.

5. Update Existing Body-Tracking Code

Currently the body-tracking code, written by Professor Eric Bachmann, can only record limb segment orientation updates to text files, and not to a network [Ref. 7]. For testing, this research parses a pre-recorded motion capture text file, and then sends update packets over the network. Modifying the body-tracking software to update to the network directly and thus eliminating file input/output could result in a significant increase in efficiency. Higher frame rates and virtual environments capable of supporting many more users would be possible.

6. Construct Avatar in Other Programming Languages

Open-source, platform-independence, and ease of authoring were major tenets of this research. Depending on the target application, programmers may have different goals. One way to meet different criteria is to construct an avatar from laser scan data in various 3D programming languages. Some possible candidates are Java3D [Ref. 32], OpenGL [Ref. 33] or DirectX [Ref. 34].

E. SUMMARY

This research has demonstrated an efficient, cost-effective method of converting laser scan data into realistic, dimensionally accurate avatars. The avatars are open-source and platform independent, and can be controlled via either programmed behaviors or by real-time network updates. Real-time avatar control was developed using the Magnetic Angular Rate Gravity (MARG) sensors developed at the Naval Postgraduate School.

LIST OF REFERENCES

1. http://www.intel.com/intel/museum/25anniv/hof/moore.htm
2. Yang, D. J., "Leaving Moore's Law in the Dust," *U. S. News & World Report,* Business & Technology, July 10, 2000.
3. TeleChoice, http://www.xdsl.com/content/resources/deployment_info.asp
4. Singhal, S. and Zyda, M., *Networked Virtual Environments: Design and Implementation*, ACM Press (SIGGRAPH Series), 1999.
5. Cyberware, http://www.cyberware.com
6. Human Animation Working Group, "Specification for a Standard Humanoid, Version 1.1.", [http://ece.uwaterloo.ca/~h-anim/spec1.1/]. August 3, 1999.
7. E. R. Bachmann, *Inertial and Magnetic Tracking of Limb Segment Orientation for Inserting Humans into Synthetic Environments*, Ph.D. Dissertation, Naval Postgraduate School, Monterey, CA, December 2000.
8. Pope, A. The SIMNET network and protocols. *Technical Report 7102.* Cambridge, MA: BBN Systems and Technologies, July 1989.
9. IEEE standard for information technology – protocols for distributed simulation applications: Entity information and interaction. IEEE Standard 1278-1993. New York: IEEE Computer Society, 1993.
10. Reece, D.A. and Dumanoir, P., "Conventions for Representing Humans in a DIS Exercise: The DWN Experience." [http://www.asset.com/orl/disaf/papers/98f-siw-245_toc.htm]. July 2000.
11. http://www.nasatech.com/NEWS/rpd999/rpd999.3dscan.html
12. http://www.laserscan3d.com/htm/technology.htm
13. http://www.wwl.co.uk/triform.htm#How it works
14. http://www.cyra.com/hdwrdesc.html
15. http://www.cyberware.com/products/wbInfo.html
16. Brutzman, D., "The Virtual Reality Modeling Language and Java," *Communications of the ACM,* v. 41(6), pp. 57-64, June 1998.

17. Ames, A.L., Nadeu, D.R., and Moreland, J.L., *VRML2.0 Sourcebook*, John Wiley & Sons, Inc, 1997.

18. VRML 97, International Specification ISO/IEC IS 14772-1, December 1997, available via www.vrml.org

19. http://h-anim.org

20. Shockley, J.W., and Morganthaler, M., "Using H-ANIM to Represent Human Figures in HLA-Compliant Simulations," *Proceedings of the Spring 2000 Simulation Interoperability Workshop*, paper 00S-SIW-061, 26-31 March 2000.

21. Livingston, M. and State A., "Magnetic Tracker Calibration for Improved Augmented Reality Registration," *Presence: Teleoperators and Virtual Environment*s, Vol. 6, No. 5, 1997, pp. 532-546.

22. http://www.animazoo.com/

23. Durlach, N., and Mavor, A., National Research Council, *Virtual Reality: Scientific and Technological Challenges*, National Academy Press, Washington, D.C., 1995.

24. Foxlin, E., "Constellation: A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications," *Proceedings of SIGGRAPH `98*, ACM, 1998, pp. 371-378.

25. Bachmann, E., McGhee, R., Whalen, R., Steven, R., Walker, R., Clynch, J., Healey, A., and Yun, X., "Evaluation of an Integrated GPS/INS System for Shallow-water AUV Navigation (SANS)," *Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology*, AUV '96, Monterey, CA, 3-6 June, 1996, pp. 268-275.

26. http://www.cyberware.com/products/decimate.html

27. http://www.aliaswavefront.com/en/WhatWeDo/maya/see/solutions/soln_intro.shtml

28. http://www.3q.com/home.asp

29. http://www.web3d.org/WorkingGroups/vrtp/dis-java-vrml/

30. Mahmoud, Q., Distributed Programming With Java, Manning Publications Company, 2000.

31. *IEEE Standard for Distributed Interactive Simulation – Application Protocols,* IEEE Standard 1278.1-1995, IEEE, Inc., 1995.

32. http://java.sun.com/products/java-media/3D/

33. http://www.opengl.org/

34. http://www.microsoft.com/directx/

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Dr. Eric Bachmann
   Kreger Hall 230H
   Department of Computer Science & Systems Analysis
   Miami University
   Oxford, Ohio 45056 USA

4. Dr. Xiaoping Yun
   Naval Postgraduate School
   Monterey, CA 93943-5101

5. Dr. Don Brutzman, Code UW/BR
   Naval Postgraduate School
   Monterey, CA 93943-5101

6. Dr. Michael J. Zyda, Code CS/Zk
   Chair, Modeling, Virtual Environments and Simulation (MOVES)
   Naval Postgraduate School
   Monterey, CA 93943-5101

7. Cindy Ballreich
   Computer Associates
   Huntington Beach, CA 92647

8. Curtis Blais
   Institute for Joint Warfare Analysis
   Naval Postgraduate School
   Monterey, CA 93943-5101

9. Andrej Cedilnik
Computer Graphics and Mobile Computing Consulting
Halethorpe, MD 21227

10. Kevin Riley
Spectrum
Tampa, FL 33602

11. Laura Dekker
3Q, Inc.
Harefield, United Kingdom

12. Brad Lawrence
United Space Alliance
Cape Canaveral, FL 32920-4304

13. Marcio Cabral
University of Sao Paulo
Sao Paulo, Brazil

14. LT Allen Dutton
921 New Mill Drive
Chesapeake, VA 23322

15. Clayton Dutton
Discovery Communications Incorporated
Bethesda, MD 20814-3579

16. David Dutton
9919 Valley Park Drive
Damascus, MD 20872

17. Susan Addleman
Cyberware Laboratories
Monterey, CA 93940

18. Alper Sinav
116 Moreel Road
Monterey, CA 93940

19. Kimberly Dutton
921 New Mill Drive
Chesapeake, VA 23322