

Directing Animated Creatures through Gesture and Speech

by

Joshua Bers

A.B., Computer Science
Dartmouth College, Hanover, NH (1993)

SUBMITTED TO THE PROGRAM IN MEDIA ARTS AND SCIENCES,
SCHOOL OF ARCHITECTURE AND PLANNING,
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE
in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1995

© Massachusetts Institute of Technology 1995

All Rights Reserved

Signature of Author _____

Program in Media Arts and Sciences
August 11, 1995

Certified by _____

Richard A. Bolt
Senior Research Scientist, MIT Media Laboratory
Thesis Supervisor

Accepted by _____

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

Chairman, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

OCT 26 1995

ARCHIVES

LIBRARIES

Directing Animated Creatures through Gesture and Speech

by

Joshua Bers

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on August 11, 1995, in partial fulfillment of the requirements for the degree of
Master of Science
in Media Arts and Sciences
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Abstract

This thesis presents a system for directing animated creatures' motions with gestures and speech. A prototype system allows the user to verbally specify a motion known to the creature and subsequently modify it using gestural input. Using human subjects, an experiment was performed to examine the ways in which a director conveys (through gestures and speech) a manner of motion to an actor. The results of this experiment motivated the design of the two prototype systems. The first system examines *direct, real-time* mapping of a user's motions onto a virtual human upper-body; the second looks at *indirect, asynchronous* mappings from the user's motions and verbal commands to a virtual bee. The architecture of the prototypes depends on four major elements which were built by the author: (1) A body model server that maintains a geometric description of the user's posture from sampled signals of sensors placed on the user; (2) A segmentation scheme for selecting activity in parts of the body; (3) A graphical interface for analyzing the sensed data and its segmentation; and (4) two animated creatures (human and bee), both with low-level joint angle control and one with additional high-level functional control (bee).

This work was supported in part by the Advanced Research Projects Agency under Rome Laboratories Contracts F30602-92-C-0141 and Thomson-CSF.


Directing Animated Creatures through Gesture and Speech

by


Joshua Bers

The following people served as readers for this thesis:

Reader _____


Alex P. Pentland
Associate Professor of Computers, Communication
and Design Technology
Program in Media Arts and Sciences

Reader _____


Marilyn A. Walker
Research Scientist
Mitsubishi Electric Research Laboratories

Contents

| | | |
|-------|--|----|
| 1 | Introduction..... | 10 |
| 1.1 | Scenario..... | 10 |
| 1.2 | Motivations..... | 11 |
| 1.2.1 | Paradigm of human-human directing..... | 11 |
| 1.2.2 | Evidence for intermediate level control..... | 12 |
| 1.3 | Example interaction with the BeeSystem..... | 14 |
| 1.4 | Summary of accomplished work | 14 |
| 1.5 | Preview | 15 |
| 2 | History and Background..... | 16 |
| 2.1 | Gesture and speech..... | 16 |
| 2.1.1 | Multi-modal interfaces..... | 18 |
| 2.2 | Animation control..... | 18 |
| 2.3 | Body models..... | 20 |
| 3 | Building Blocks | 22 |
| 3.1 | Speech recognition..... | 23 |
| 3.2 | The BMS system..... | 23 |

| | | |
|---------|---|----|
| 3.2.1 | Device Layer | 24 |
| 3.2.2 | The Body Model Layer | 26 |
| 3.2.2 a | Structural layout | 26 |
| 3.2.2 b | Data format layout..... | 27 |
| 3.2.2 c | Joints: the model's flexible glue..... | 29 |
| 3.2.2 d | Joint angle computations..... | 31 |
| 3.2.2 e | Calibration of the model | 33 |
| 3.2.2 f | The update cycle..... | 34 |
| 3.2.3 | Network Server Layer..... | 34 |
| 3.3 | Animation system..... | 36 |
| 3.3.1 | Joint angle control of an articulated hierarchy (joint-segments) | 37 |
| 3.3.2 | High-level functional control: motor programs..... | 37 |
| 3.3.3 | Integrating motor and direct control..... | 38 |
| 3.4 | Visualization of body data | 38 |
| 3.4.1 | Matlab Body Motion Analysis Interface..... | 39 |
| 3.4.2 | Segmentation and selection of body motions | 40 |
| 4 | The Working Prototype Systems..... | 43 |

| | | |
|-------------|--|----|
| 4.1 | Direct Motion Mapping System (DMMS)..... | 44 |
| 4.2 | Interpretative motion mapping: The BeeSystem..... | 47 |
| 4.2.1 | Segmentation of body data..... | 47 |
| 4.2.2 | The bee and its motor programs..... | 49 |
| 4.2.3 | Mapping from input to output..... | 50 |
| 4.3 | Summary..... | 51 |
| 5 | The Next Step | 53 |
| 5.1 | Multiple dynamic commands..... | 54 |
| 5.2 | Static posture control of Bee..... | 55 |
| 5.3 | Allow speech modifiers..... | 55 |
| 5.4 | Synchronization of motor and director control..... | 55 |
| 5.5 | Conclusion | 56 |
| Appendix A. | Director-actor Pilot Experiment..... | 57 |
| A.1 | Experimental Protocol..... | 58 |
| A.1.1 | Experimental Setup:..... | 58 |
| A.1.2 | Procedure Description:..... | 58 |
| A.1.3 | Instructions: | 59 |
| A.2 | Results and observations of experiment..... | 60 |

Appendix B. Data Format for BMS Clients63

Appendix C. Joint Angle Calculations for the Body Model.....67

 C.1 Hand and wrist.....67

 C.2 Elbow68

 C.3 Shoulder69

 C.4 Eye70

References.....71

Acknowledgments

I would like to thank my advisor, Richard Bolt, for pointing me towards the body model work. The *aardvarks* who made lab-life more fun and productive: Sara Elo, Jeff Herman, Earl Rennison, Deb Roy and Nicolas St-Arnaud. My readers Sandy Pentland and Marilyn Walker for their guidance on drafts of both my proposal and this document. My third reader who has provided invaluable comments and support; I love you dad. The UROP crew: Alberto Castillo, Mischa Davidson, and especially Juno Choe for their help with the body model code. Jen Jaccoby, Tomoko Koda, Obed Torres and specially my officemate Kris Thorisson who all helped make life in AHIG bearable. Justine Cassell has my thanks for helping me classify the gestures from the experiment. Linda Peterson for the stability of good advice.

Mom and dad for life.

Marina for love.

Notices

The following trademarks are referred to throughout this document:

- Hark is a registered trademark of BBN Hark Systems a division of Bolt Beranek and Newman, Inc.
- Starbase is a registered trademark of Hewlett Packard.
- *CyberGlove* is a registered trademark of Virtual Technologies.
- Flock of Birds is a registered trademark of Ascension Technology Corporation
- Matlab is a registered trademark of the Math Works Inc.

1 Introduction

To animate: to give life and soul to a design, not through the copying but through the transformation of reality.¹

1.1 Scenario

Imagine that you have created a virtual bee that knows how to fly and now you want to animate it. You sit down at your computer to block out the action for the sequence.

A rendered image of the bee appears on the screen on a virtual 3D stage. It flies around the stage in a circle in a normal manner. “Flap your wings slower...” you say, and with both of your arms you demonstrate a few flapping movements. Your hands slowly spread upwards and then reach a peak before descending at a slightly faster pace than on the up-stroke.

You look up to the screen to see your creature flying as before except this time with a wing-flap that looks a lot like the example that you gave. You change your mind and say, “No, I want you to fly faster.” The creature begins again, the style of flight is the same but its velocity has increased.

In everyday conversation we understand what someone means whether they use their whole body to describe a motion or just their hand. A natural interface based on a conversational model would handle both kinds of descriptions.

¹Collective statement of cartoonists of the Zagreb studio[1]

This thesis has as its "holy grail" the above scenario. The work described herein consists of building blocks and prototype systems that aspire towards this goal. Specifically, I have developed: a sensor based model of the user's body for real-time capture of body motions; a prototype system that maps from the input model to a 3D rendered graphical human body; a second prototype that interpretively maps from a user's speech commands and gestures onto animated actions by a virtual bee.

1.2 Motivations

1.2.1 Paradigm of human-human directing

Imagine a stage director explaining to actors how they should walk down stage. How will he tell them what to do? He will give them the flavor of the walk with a brief demonstration with his own body. The body is a great descriptive tool that is constantly at our disposal. Animators, directors, dancers and others use the body to describe and create motions in bodies. The goal of this thesis is to demonstrate a computer system that enables the user to modulate the motions of an animated creature using his body and speech.

In human-human directing, most of the time we do not control the actor with puppet strings nor would we want to. Instead, we use verbal descriptions, e.g., "walk as if you are drunk." Then, it is up to the actor to interpret our commands. Sometimes, however, we would like the fine control of puppet strings, i.e., "walk this way," followed by an example of walking by the director. Here the actor must extract the features from the visual example that the director is conveying and then integrate these with his default walking behavior.

1.2.2 Evidence for intermediate level control

In order to investigate human-human director-actor interactions I conducted a small experiment. The task was for a director to convey a style of walking motion to the actor using only his upper body and speech. A complete description of the setup and results appear in Appendix A. Consider the following example below:

Director: "I want to you to walk over there." [Index finger sweeps from left to right.]

Actor: [Walks across the room with arms hanging straight down.]

Director: "Your arms are going like this," [swings his index and middle finger]. "I'd like them to be more folded." [swings fingers bent at first joint. Then switches to swinging both arms from the shoulders bent at the elbow].

The design of the second prototype system presented in this thesis was motivated by three features of the above interaction, taken from a pilot experiment (Appendix A): (1) speech is used to select an action: walking; (2) periodic gestures indicate manner of motion for a specific body part; (3) the gesture may switch from an observer viewpoint to a character viewpoint, e.g. changing from swinging the fingers to swinging the arms.

In a large percentage of the interactions in the experiment, the director began with a high-level verbal description of the motion/action to be carried out. This was usually followed by a modifier clause consisting of speech and gesture that pinpointed an important feature of the motion.

Analysis of the data showed that in 34 out of 40 gestures that co-occurred with speech, the director's gesture indicated a manner-of-motion; these were

mostly periodic. Half of the gestures were from the character viewpoint and half from observer viewpoint (see section 2.1). These results are similar to those from a previous study of the iconic gestures of people narrating a cartoon they had seen, which noted that 60% had an observer viewpoint and 40% a character viewpoint [2]. According to McNeill, the observer viewpoint tends to exclude the speaker's body, forcing the hands to play the role of the character[3]. My observation was that the director would switch from one viewpoint to another to gain different perspective on the motion being described. The constraints on director's gesture space (camera's field of view) may have inhibited the use of the full body and thus the character viewpoint.

The question may arise as to why one needs to use both speech and gestures rather than, for example, gesture alone. By themselves, speech and gestures can be ambiguous. For example, if one gives the command, "walk," one has left out how the arms should move, the speed, etc. all of which could be specified through one gestural example. On the other hand, a gesture without the context of speech is very difficult to interpret. When combined, they reinforce the interpretation of the other, e.g. in "fly like this," the speech helps give the semantic and temporal context for the gesture (what it represents and when it occurs) and the gesture provides the manner of the motion indicated in speech (how it should be done).

The first working prototype realized in this thesis, mapped human body movements directly onto those of a 3D human body model. Providing all of the motion for the animation was quite tedious and motivated the second prototype. The second system (BeeSystem) explored indirect control of the flight of a graphical bee. This included selection of high-level functional motion through speech and low-level joint angle control through mapping

of segmented gestural examples. I chose the flight of a simplified bee as my domain of motion. This reduces the complexity from animating a human walk (with all of its degrees of freedom and expectation of realism evoked in the viewer) to that of a parameterized bird flight with only 3 degrees of freedom (wing flap, vertical bounce, forward speed).

1.3 Example interaction with the BeeSystem

Consider the example interaction with the BeeSystem below:

User: "Fly."

System: Bee flies around in a circle above a stage.

User: "Fly like this." (makes flapping motions with his fingers)

System: Bee flies around stage flapping its wings in same manner as in the user's example.

User: "Fly like this." (makes flapping motion with whole arms)

System: Bee flies around with the flapping dynamics provided from the example.

While being simplified from the experimental setup of 1.2.2, this example retains the three key features mentioned above: (1) speech to select action (2) periodic gestures (3) switch in viewpoint of gesture.

1.4 Summary of accomplished work

The major accomplishments of this work are: 1) the implementation of a device independent data-layer for capturing and representing the body posture of the user; 2) a graphical user interface in Matlab for analyzing and segmenting movements captured with 1; 3) an interface that enables a real-time mapping from the movements of the user onto a graphical upper body;

4) an interface that combines speech and segmented gestures to control the flight of a graphical bee through both motor and direct control.

1.5 Preview

- Chapter 2 reviews relevant work from related research areas.
- Chapter 3 presents the building blocks of the thesis: speech, creature animation and body sensing.
- Chapter 4 discusses the two working prototypes: the direct and interpretative motion mapping systems.
- Chapter 5 details some enhancements to the current prototypes that would be the next step for this research.

2 History and Background

In this chapter I will review some relevant research from the following three areas: Gesture and speech in human-human as well as human-computer interactions; Animation control; and Human body modeling.

2.1 Gesture and speech

Research on gesture and speech in human communication is extensive. See [3-5] for good background. The most common gestures that appear in everyday conversation are[6]:

- **Iconics:** enact the semantic content of the speech, e.g. The left hand sweeps across the body and forward as the speaker says, "[he ran out of the room.]²"
- **Metaphorics:** portray the speaker's abstract ideas about what is being said; e.g. the speaker's hands are cupped together in front of his chest; then they open up suddenly and spread apart, while saying, "then he went [crazy]." The gesture exposes the speaker's connection of going crazy to an explosion.
- **Beats/Butterworths:** mark the rhythm of speech. The speaker holds hand in a fist and shakes it periodically while saying, "[we need to stay the course and follow the thousand points of light.]"
- **Deictics:** these pointing gestures indicate objects around the speaker.

²Brackets indicate speech that co-occurs with gestures by the speaker.

The iconic gesture type is of interest to this thesis. When directing body movements of an animated creature we enact how we want that creature to move. Our body or body parts embody the concrete motion that we are trying to convey.

There exists another distinction within the iconic category of gesture, which is one of viewpoint. There are two possible viewpoints: character or observer. Character viewpoint corresponds to the speaker taking an active role and becoming the subject of the speech.³ Observer viewpoint gestures are those where the speaker is indirectly indicating the action in his gestures. For example, take the sentence from the experiment, presented in 1.2.2, where the director says, "Your arms are going like this," while swinging his index and middle finger (observer viewpoint). "I'd like them to be more folded," he says as he swings his fingers bent at the first joint (observer viewpoint). Then he switches to swinging both arms from the shoulders with the elbows bent.

Given the goal of directing a creature what kinds of gestures do we need to look for. The experiment in Appendix A showed that there are a few kinds of gestures involved in directing motions: those that convey the manner of motion; those that indicate the body part involved; and those that show the directionality of the motion [7]. Because the movement domain, walking, was periodic, the manner-of-motion gestures observed in the experiment tended to be periodic, the same can be expected of gestures depicting the flapping motion of a bee in flight.

³Rimé and Sciaratura label character viewpoint as pantomimic gestures [5].

2.1.1 Multi-modal interfaces

Some authors have supported the integration of multiple modalities into computer interfaces [8-10]. A few have built prototype systems combining pointing devices (mouse, touch-pad, pen) and natural language (speech or typed input) [11-13], and even fewer have built systems that combine speech with natural gesture[14, 15]. The "Put-that-there" system allowed the creation of graphical objects and their placement on the screen using speech and deictic gestures of the hand[16].

Koons et. al. built a system (ICONIC) that recognized iconic gestures when combined with speech from the user to facilitate the placement and movement of objects in a virtual room[14]. One example from ICONIC illuminates the differences with the current work. In the demonstration video of the ICONIC system a virtual mouse was placed on the floor of the virtual room and the user said, "[move it this way]" while sweeping his left hand across his chest from left to right (observer-viewpoint)[17]. Here the gesture was referring to the path for the object (directionality).

This thesis is concerned with giving the user control of the *manner of motion* through both character and observer viewpoint iconic gestures.

2.2 Animation control

At the opposite ends of the spectrum of animation control are direct control at the structural level, and guiding at the goal or task level [18]. Below we review previous work of both of these; the first two are examples of direct control and the following two illustrate goal or task level control.

Some of the earliest work in the domain of body-sensing for driving animation was the scripting-by-enactment/graphical marionette work of Ginsberg and Maxwell [19-21]. They made use of biangulated cameras to track L.E.D.'s placed at key locations on the body of the user. With the help of some image processing and trigonometry, they could extract the location of body parts in three dimensions. They then extracted joint angles from their model and used them to guide a human figure animation system.

In the film industry, the use of body-sensing devices to map directly from the user's body onto an animated figure is known as *performance animation* or *motion capture*. The TV cartoon character *Moxy* is animated in this way.

Direct control of creatures, however, does not always yield a desirable effect. Frequently a professional is needed to achieve the desired look. As famous animator Shamus Culhane points out, "the best use of animation is when it caricatures, not imitates, real life" [22]. Tediousness and repetition are some more drawbacks of having to do everything yourself. The resulting behavior of these systems depends heavily on the physical abilities of the user. Another drawback of direct control is that it does not allow for control of a body with different articulation from one's own. Thus a handicapped person could not make full use of such a system and more generally users are limited in the diversity of motions and creatures controllable.

Badler et al. have looked at controlling animation through a natural language interface [23]. They built a system that could handle commands such as "roll the ball across the table," by representing the geometric, kinematic and the dynamic properties of actions.

The ALIVE project at the Media Lab allows full body interaction with autonomous creatures. The level of control over the creatures is behavioral; through camera based recognition of the user's gross body movements the creatures change their goals[24]. For example, if you wave, the dog will walk toward you. Here, gestures are treated as symbolic units.

2.3 Body models

The process of representing human body motion in a computer involves modeling the geometry of the body, and sampling motions of its parts. In the following we summarize previous body modeling systems and the issues of sampling real motions and segmenting them.

The idea and implementation of body models are not new. Computer body models have been used extensively in computer graphics modeling systems, animation, and in human factors research[19, 23, 25, 26]. In chapter 2 of [26], Korein surveys early work on body models used for ergonomic modeling and graphical simulation. Zeltzer did some early work with body models for realistic computer graphics simulation of a walking skeleton [27]. Ginsberg and Maxwell used a hierarchical body model as part of a key-framed animation system that used optical body tracking [19, 20]. In "Virtual Sailor," Chen et al. used a body model to control a virtual person who could respond to gestures from a user in real-time [28]. More recently Badler et al. [23] have created a complex human model, *Jack*TM. In chapter 2 of [23], Badler et al. provides a thorough history of research in the area of body modeling and the study of human motion.

The tradeoff in all sensor-based modeling systems is between how much of the modeling is done by the system and how much information is sampled

from the environment.⁴ Most joint based models that use sensors to track a user in real-time utilize the data to drive complex, constraint-based inverse kinematics models [30, 31].

Accuracy of a body motion sensing system means a spatio-temporal match with reality. No matter how spatially accurate (close to reality) a body model is, it is useless unless it samples the sensors at a sufficient rate. According to Fitts and Van Cott, a human can exhibit internally generated trajectories and responses with up to 5 Hz components [32, 33]. Reflex actions and isometric responses can contain up to 10 Hz components [34, 35]. Eye fixations last on average 180-300 milliseconds (3-5 Hz) [36]. Thus a 20-25 Hz sampling rate is sufficient to capture all of these types of body motion.

⁴For example, in a system that uses visual tracking clues, the system would need to compensate for imprecise sensing by using a model with more constraints and error correcting feedback loops (see [29]).

3 Building Blocks

Before multi-modal interaction with a computer system is feasible the computer must be able to see and hear and produce output to the user. This chapter describes the components of a perceptual level that serves as the foundation for the BeeSystem.

The building blocks of the system include a body model of the user's body, a speech recognizer and an animation system (Mverse). The body model server was specified by the AHIG⁵ and was implemented by the author, the speech recognizer is a commercial product from BBN (Hark), and the graphics system Mverse was developed in-house by Chris Wren.

The Body Model Server (BMS) gives "eyes" to the system by maintaining information about the user's body position and posture. It gathers data from magnetic position and orientation sensors, data gloves and the eye tracker worn by the user. This setup is not ideal because the wires attached to the sensors encumber the user and restrict his range of movement. However, it eliminates the problems of occlusion and background noise that face vision based tracking systems such as in Pentland et al. [24]. The other main function of the BMS is to decouple the higher perceptual layers i.e., clients, from the specific sensing hardware used. Thus, eventually, this will allow us to replace the sensors with newer technology.

⁵Advanced Human Interface Group at the MIT Media Lab: Richard Bolt, David Koons, Alan Wexelblat, Kris Thorisson.

3.1 Speech recognition

The function of speech in this thesis is twofold: command and gesture selection. The user's speech explicitly selects a command. For example if the user says "fly like this," word spotting reports the command "fly". Gesture selection is done by looking for a gesture that is temporally synchronous with the user's utterance.

Speech recognition is accomplished by the Hark® system.

3.2 The BMS system

The BMS is composed of three modular units or layers (see Figure 3.1):

The Device layer:

Handles data acquisition from the sensing hardware. It provides an interface to the data and devices for the *Body Model* and the *Server* layers.

The Body Model layer:

Contains the geometric representation of the model. It defines the objects that make up the model: joints, segments, and their specific instances: head, arm, shoulder, etc. It also defines the methods to calculate joint angles and provides structured access to the body data for the *Server*.

The Network Server layer:

Functions as a gopher between clients and the Body Model. It maintains information about each client including parts requested and data rate so that it can ship the requested data at the right frequency.

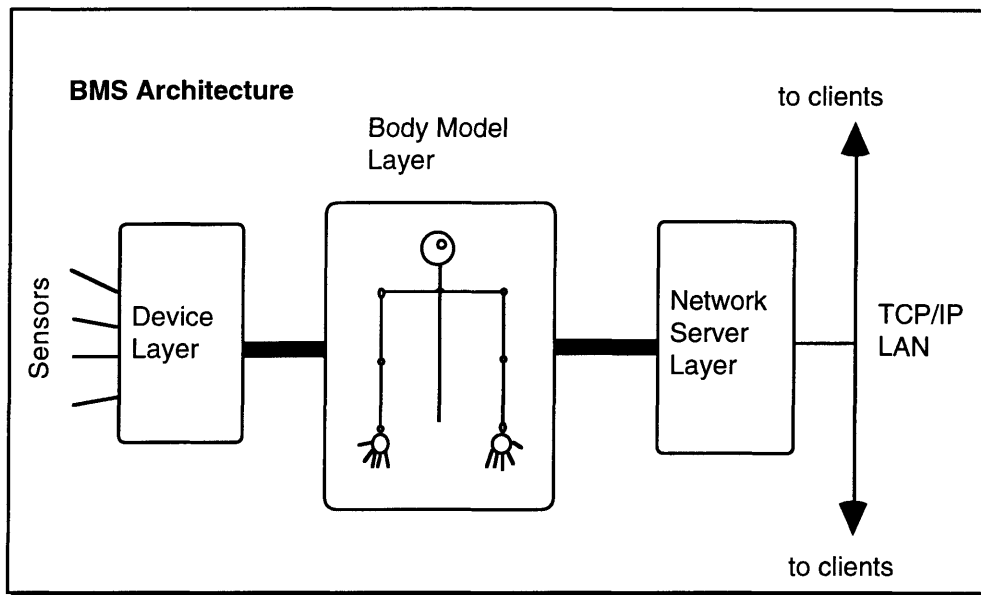


Figure 3.1: The three layers of abstraction that insulate clients from the body-sensing devices.

The Body Model and Server layers are written entirely in C++. The device layer is a combination of C++ and assembly code. The code reflects this modular decomposition in its major class definitions. There is a device class, a body class, and a server class. One instance of each is created to form the BMS.

3.2.1 Device Layer

The device layer is mainly concerned with reading data from the device streams and packaging it into formats readable by the body model. There are three kinds of data structures that the body model expects from the devices: 4x4 transformation matrices from the 4 six-degrees-of-freedom (DOF) sensors

(Flock of Birds); special data records that holds the 18 sensor values from the gloves; and the horizontal and vertical offset of the pupil values from the eye tracker.

The data output from all of the devices is fed into a specialized multi-port serial communications card in a '486-DX2 PC running at 66 MHz. This card (an Arnet Smartport™ with 8 RS-232 ports) buffers each data stream, leaving the main processor free for other tasks. This enables parallel and asynchronous capture of all the data streams without regard for their differing data rates: 100 Hz for the birds, 85 Hz for the gloves and 60 Hz for the eyes. Subsequent filtering of the buffered data can then be performed by the main processor when it is ready to sample the devices again. Noise is filtered from the birds through averaging of the accumulated data, however, a 3 point median filter can be substituted or added on top of the averaging to help with sensor spike removal.

The device layer also performs basic utility functions for synchronizing the PC's clock with a reference, aligning sensors, adjusting data filters, and getting time-stamps. Every time the device data are updated the device layer sets the time-stamp with the current time in 1/100th of seconds since midnight. This information is useful to clients who wish to synchronize with other time-dependent processes such as a speech recognizer [14].

The next section discusses the structure of the body model layer.

3.2.2 The Body Model Layer

3.2.2 a Structural layout

Foley et al. point out three main purposes of geometric models: to convey a spatial layout; to show the connectivity of components; and to associate data-values with specific components [37]. Although all three apply, the last is the main purpose of the geometric model in the BMS. We chose a hierarchical, geometric model as the representation system because of three desirable features: a logical way of dividing up the body; efficient computation; and easy extensibility.

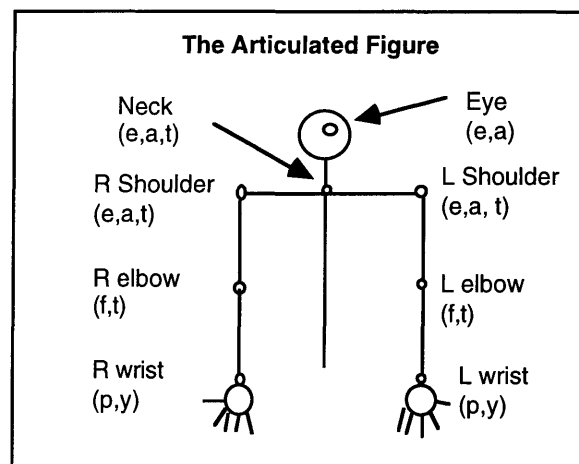


Figure 3.2: The upper body is hierarchically divided into joints and segments that make up the articulated figure. Here, each joint is named and its degrees of freedom are shown in parens (e=elevation; a = abduction; t = twist; f = flex; p = pitch; y = yaw). The root of the hierarchy is the torso with the distal segments as branches and leaves.

The upper body can be thought of as a tree with the torso as the root and the arms and head as the branches (see Figure 3.2). Motion in proximal segments propagates to their distal segments; the leaves on a branch move with it. The hierarchy is constructed by attaching each segment to its proximal and distal neighbors through a joint. The joint-segment complex that maintains the

spatial relationship between adjacent segments is discussed in sub-section 3.2.2(c).

The object classes of our model are defined in terms of multiply occurring segments or joints such as fingers, arms, and elbows. This building-block approach makes it easy to extend the model. Adding new parts involves defining new derived classes of the segment and joint super-classes and defining how and where they connect with the rest of the model.

Now let's look at the data format and representational conventions used in the model before examining the joints.

3.2.2 b Data format layout

The internal representation for body data follows the conventional translated coordinate frames used by the robotics and computer graphics communities [38]. The external or client format measures joint angles relative to a relaxed standing pose (see Appendix B for the detailed specification). All position values are in centimeters and angular values are in degrees.

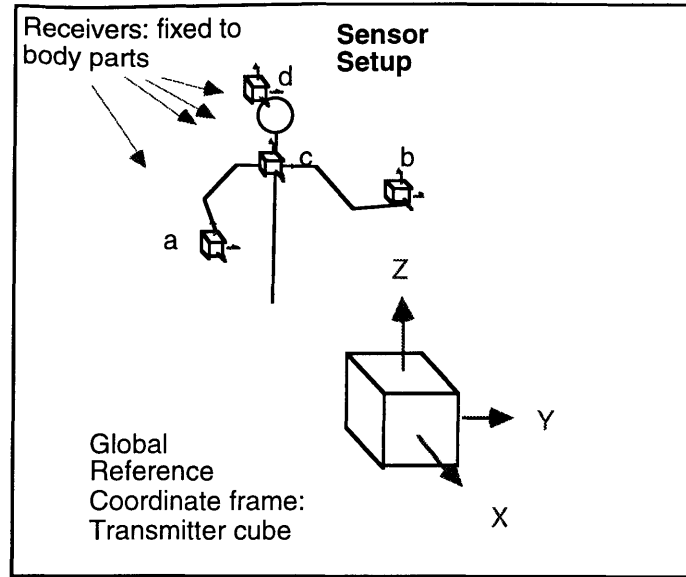


Figure 3.3: Bird® sensors are placed to landmarks on the body. One is affixed to the back of each *CyberGlove*® on the cuff above the wrist (a,b), one is located near the base of the neck, attached to the liner in the back of a jacket worn by the user (c), and one is attached to the eye-tracker, worn on the head (d).

The transmitter cube defines the origin of our global coordinate system (see Figure 3.3). Figure 3.4 shows how the position and orientation of a body segment are represented internally with a 4x4 matrix. The last column of the matrix gives the translation from the origin to the segment's proximal end. The orientation is given by three orthogonal vectors: normal (N), orientation (O) and approach (A) found in columns 1-3 of the matrix T of figure 3.4. The use of a global coordinate system allows for interaction among objects or agents that share the same physical or virtual space. For example, two users may wish to interact with each other and with a shared screen in a virtual environment.

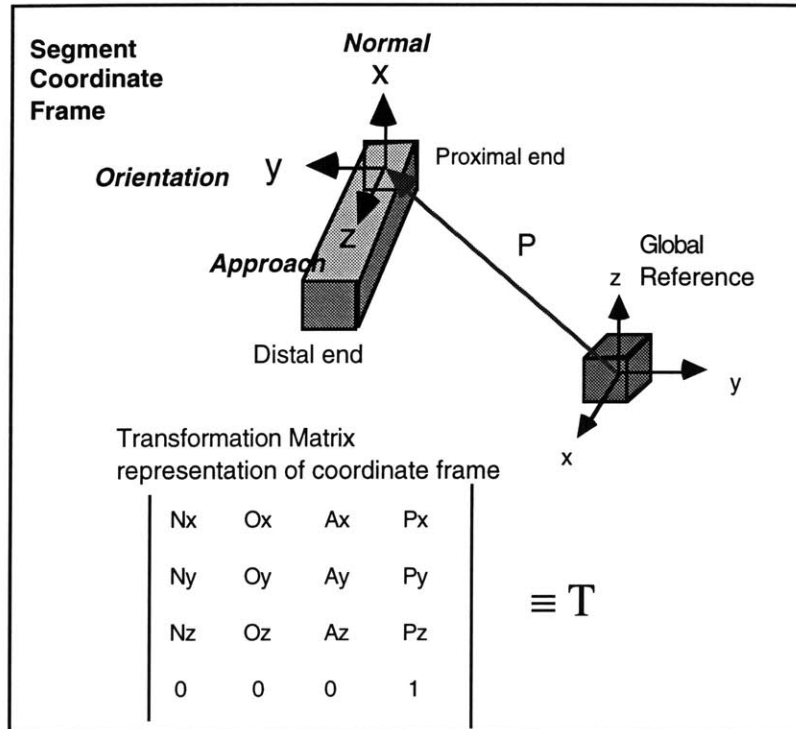


Figure 3.4: A local coordinate frame is attached to the proximal end of each segment (closest to the root of the body trunk). The frame defines the position and orientation of the segment with respect to the global frame. Below is the mathematical representation of a coordinate frame. The four vectors: N, O, A, and P correspond to the three orthonormal axes, and the position of the segment[38].

3.2.2 c Joints: the model's flexible glue

Whereas traditional Denavit and Hartenberg notation represents joints as one degree of freedom prismatic or revolute joints, the body has joints with multiple degrees of freedom such as the shoulder with spherical motion [39]. Table I shows the various combinations of motions found in the joints of our model; they are: flexion, spherical, and twist (explained in the caption of Table 3.1). These motions have one, two and one degrees of freedom respectively. The joint angles that we model are those of the hands (15x2), the wrists(2x2), the elbows(2x2), the shoulders(3x2), the neck(3) and the right eye(2); namely, 49 degrees of freedom to represent the upper body posture.

Table 3.1: Joint Types and their motions, degrees of freedom and rotation axes. Axes of rotation are fixed to the proximal segment except for twist (shown in bold-face), which represents axial motion or rotation about the z axis of the distal segment. Flexion works much like a hinge that opens and closes about a fixed axis. Spherical motion is movement along both the longitude and latitude of a virtual sphere centered on the joint.

| Joint | Types of motion; DOF | Axes of Rotation |
|--------------|-----------------------------|-------------------------|
| Shoulder | spherical, twist; 3 | Y, Z, Z |
| Elbow | flexion, twist; 2 | Y, Z |
| Wrist | spherical; 2 | X, Y |
| Fingers | flexion; 1 | Y |
| Neck | spherical, twist; 3 | Y, Z, Z |
| Eye | spherical; 2 | Z, Y |

Figure 3.5 shows one way to conceptualize a generic joint: the joint-segment unit. The joint-segment is a joint with its attached proximal and distal segments. The joint consists of two spheres, each with its own set of axes or coordinate frame. One frame (the moving frame) rotates within the other (the fixed frame). The distal segment is fixed to the moving coordinate frame and the fixed frame is attached to the proximal segment by a static relation. All joint angles, except for twist, specify rotations of the distal segment relative to the proximal segment. Twist is a rotation of the distal segment about its own z-axis.

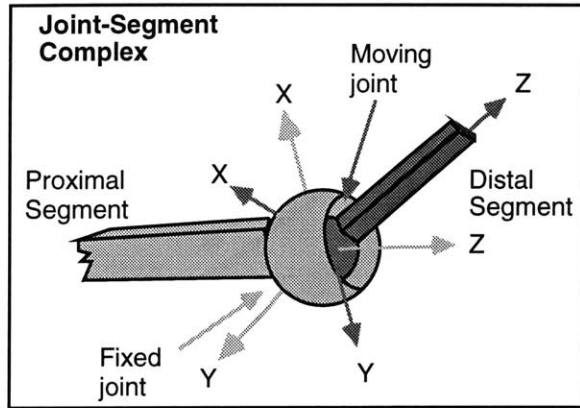


Figure 3.5: The joint-segment complex contains the fixed and moving coordinate frames of a joint. The representation is based on Korein's segment connection method. Mathematically, if T_p is the proximal segment's coordinate frame then the distal segment's coordinate frame $T_d = T_p * T_j$ where T_j is the rotation matrix given by the joint angles.

3.2.2 d Joint angle computations

Transform Equations. The calculation of joint angles in the body model is a two stage process. The first stage finds the transformation across a joint T_j . Transform equations are used to find this transformation. This technique adopted from Paul [22], solves for coordinate frames that are described in two or more ways [38]. As an example of transform equations, Figure 3.6 shows the shoulder and elbow defined with respect to the global reference frame. There is also a transform between the shoulder and the elbow, forming a loop or a directed transform graph. A transform equation for this setup would look like this $T_e = T_s T_{s_e}$. One could solve for T_{s_e} by pre-multiplying both sides by T_s^{-1} , or one could use the perhaps more intuitive technique of traversing the transform graph. In the latter approach, one traverses the graph from the tail of the unknown transform to its head, writing down each transform traversed from tail to head, and writing down the inverse of each traversed from head to tail.

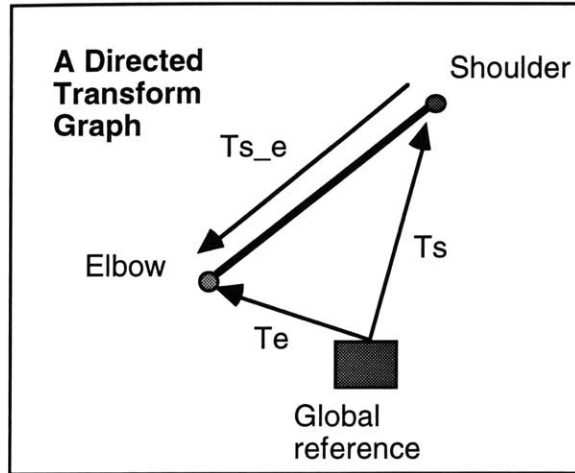


Figure 3.6: The vectors represent 4x4 transforms. Walking the transform graph to solve for T_{s_e} gives: $T_{s_e} = T_s^{-1} T_e$. See text for details.

The second phase of the joint angle calculation process converts the transformation matrices along a chain of joints and segments into joint angles. This operation can be fairly straightforward or very complex depending on how much freedom the chain has. The more one knows about the position and orientation of links in a joint chain, the easier it is to invert a transformation to get back joint angles. Paul has shown how Euler angles can be extracted from a 4x4 homogenous transformation matrix using six multiplications, three additions and three transcendental calls [38]. By contrast, a differential solution for a 6-DOF joint chain with six one DOF joints require 78 multiplications and 47 additions to get the $\text{diff}(\text{joint angles})$ from $\text{diff}(\text{position})$ (see [38, 40] for details).

In our sensing setup all the positions of the joints modeled are known in 3D space, as well as the segments' orientations that connect them, allowing

closed form solutions to all joint angles in the model.⁶ (See Appendix C for detailed descriptions of joint angle computation for our model.)

3.2.2 e Calibration of the model

To prepare the model for a user, the person must go through some initialization routines. There is a calibration phase (for each glove, for the body, and for the eye-tracker) that adjust the model to the current user's body dimensions.

The Gloves and Body. Ten calibration poses for each glove establish the ranges of its 18 sensors. The arm span is sampled from the spread of the wrist sensors during calibration. The span is scaled according to average proportions of body parts to set the lengths of all the model's segments [41]. The calibration of the trunk sensor is done by aligning its axes to those of the transmitter cube when the user is made to stand upright facing away from the cube, in its x-direction, and with his shoulders parallel to the y-axis. The head sensor is aligned in a similar manner (refer to Figure 3.3). This procedure allows for accurate measurement of angles without requiring exact placement of the sensors on the body every time the suit and eye-tracker are put on.

The Eye. For the calibration of the eye-tracker we use the techniques of Koons and Thorisson [42].

⁶To simulate our setup, grab hold of the forearm just above the wrist. All degrees of freedom in the arm except those in the wrist are lost, thus pinning down the shoulder, elbow and wrist in space. The location of the shoulder is simplified to always be a fixed perpendicular distance from the trunk (determined at calibration).

3.2.2 f The update cycle

Every time a new set of information is prepared for shipment to a client, the server must update the internal body model structures. An "update joint angles" method fills in the new joint angles using the methods described in Appendix C. Another method, "update transforms", maintains the joints' transformation matrices.

3.2.3 Network Server Layer

The client-server interface supports an environment in which different architecture machines and different programming languages need access to the BMS data. Binary integer data was chosen as the representational format over the network because of its universality and inherent size and speed advantages as compared with ASCII or floating point formats. Two-byte signed integers are used for all position, joint angle, and vector information, and four byte unsigned integers are used to represent the time-stamps. All floating point numbers are biased by a fixed amount before conversion to an integer value so as not to lose precision.⁷

The client connects to the BMS port on the server machine and indicates: (a) which body parts the client is interested in, by sending a bit-mask; and (b) at what frequency to sample the devices, by sending a data-rate value. Figure 3.7 shows the masked-out body divisions that a client may select. The server operates in two different data-modes: stream (asynchronous) or polled (synchronous). Stream mode outputs time-stamped data-records at a rate set

⁷Unit vector components are biased by 10^4 whereas angular and position values are biased by 10 before transmission.

by the client upon connection. Polled mode sends data to a client only when requested. A C++ client class hierarchy has been developed to provide transparent network access to the data at various levels of abstraction.⁸

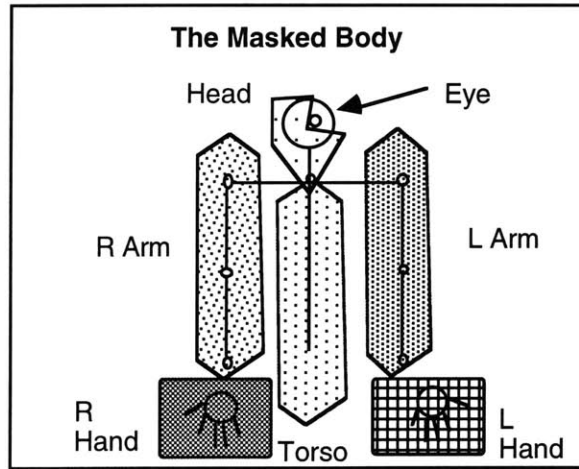


Figure 3.7: The body is divided into regions which the client can request. The names correspond to shaded areas which contain the positions and joint angles data for all of the joints and segments within the region.

At the beginning of a data transfer session, the server sends the client a set of calibration values. The calibration data consists of the minimum and maximum values for the requested joint angles and positions. The angular ranges for the model are fixed; they were obtained from human factors research for an average male (see Figure 3.8) [43, 44]. For example the elbow joint calibration is min: 0 and max: 138 for flexion, and min: -65, max: 107 for twist. Knowledge of these ranges is useful for clients who wish to compute the magnitude of motion in a joint and for thresholding or filtering purposes.

⁸Direct sample-by-sample data is the lowest level, and data filtered for features (eye fixations, nods of the head, etc.) is currently the highest.

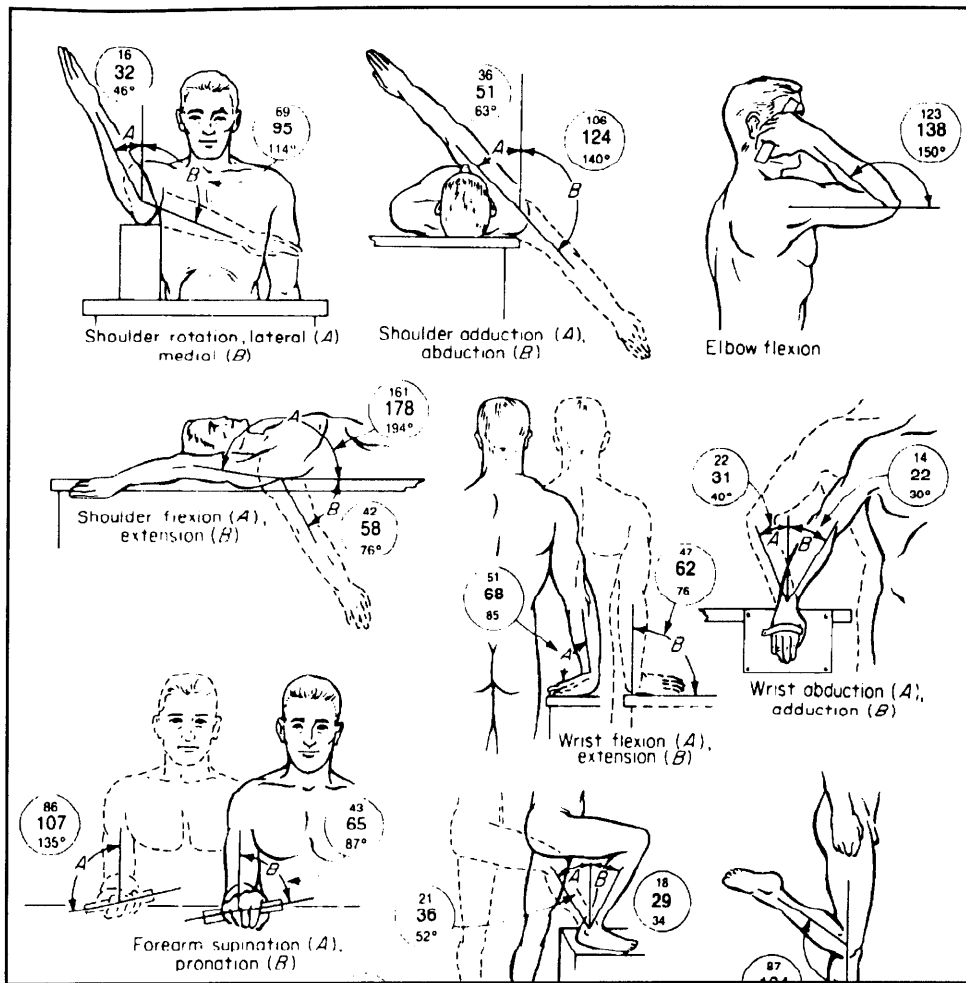


Figure 3.8: Range (in degrees) of rotation and movement of certain upper and lower extremities, based on a sample of 100 male college students. This is used to establish ranges for the joint angles in the BMS (Houy, 1983). (Reprinted with permission from *Proceedings of the Human Factors Society 27th Annual Meeting*, 1983. Copyright 1983 by the Human Factors and Ergonomics Society. All rights reserved.)

3.3 Animation system

The lowest level of the animation system is Mverse. It is a 3-d graphical object generating system; the objects are the various parts (spheres, cubes, etc.) needed to construct the animated body model and bee. Two higher levels of abstraction and control were built on top of the Mverse graphics system: posture (joint-angle) control of an articulated hierarchy; and a functional, or

motor control, system for coordinated movement which is built on top of the posture control.

3.3.1 Joint angle control of an articulated hierarchy (joint-segments)

Objects from the Mverse system are put together in units of one joint and one segment to form the joint-segment. These units can be combined, through attachment to each other, to make articulated hierarchies. The posture of such hierarchies is controlled through the joint-segment abstraction (see 2.2.2(c)). For example in the graphical body (GBody) each joint-segment provides a function called "update-joint-angles" through which rotations are applied to the joint's degrees of freedom in the appropriate order. Thus a joint with two degrees of freedom would take two parameters. Zeltzer calls this the structural level of abstraction [18]. At this level no constraints are imposed on the values of the angular parameters.

3.3.2 High-level functional control: motor programs

The highest layer of abstraction provides functional control of coordinated repetitive motion. At this level there is no direct access to the joint-segments (i.e., their degrees of freedom) instead they are indirectly controlled through the parameters of the motor programs.

Motor programs associate procedural elements (motors) with structural elements (joints) [18, 25]. These functional units take care of the details of animating a repetitive motion by cyclically driving the degrees-of-freedom of the motion through the appropriate range of motion. All motor programs take a duration parameter which indicates how many times to cycle through

the motion. Motor program execution is handled through Mverse's animation system.

The motor functions take normalized parameter values from 0 to 1, freeing the caller from any knowledge of the actual range of the degree-of-freedom involved [25]. For example in the BeeSystem *flying* is a motor program that has three parameters: frequency, flap-range, and bounce height . If the caller specifies a value of 0.5 for all parameters then the resulting motion will be an average flight. Namely the wings will be driven through half of their range, their flapping rate will be the median, and the bee will bounce half of the maximum height.

The important difference between control at the above levels of abstraction is that the first is a direct-driving one-to-one joint manipulation, while in the second a one-call-equals-many movements, indirect mapping.

3.3.3 Integrating motor and direct control

To handle the timing and synchronization of the motor controlled animation with the direct manipulation we used a special synchronous animation mode. In this mode the process performing direct joint manipulation tells the Mverse system when to cycle its animation and update the display.

3.4 Visualization of body data

The original goal of the perceptual part of the system was to extract features from periodic movements: frequency, amplitude, phase, etc. The cumbersome techniques required for such analysis (FFT, autocorrelation) combined with the loss of the subtlety of movement argued for pure segmentation and extraction of motion.

3.4.1 Matlab Body Motion Analysis Interface

Using Matlab software, a visualization tool was built to determine how to segment data from the BMS into regions of activity.

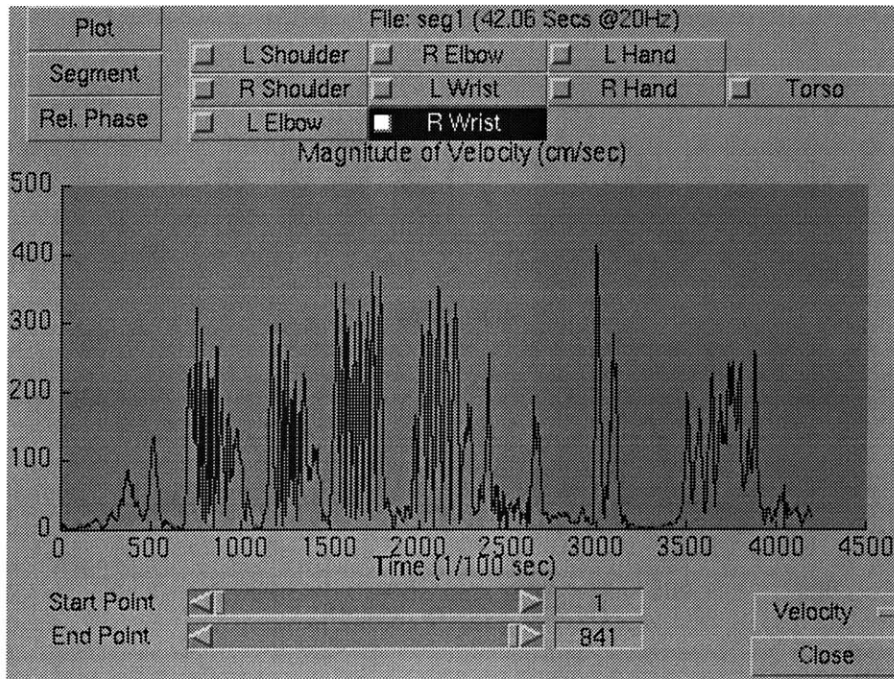


Figure 3.9: The Matlab analyzer interface window.

The Matlab analyzer has the following capabilities:

- Plotting the magnitude of velocity or kinetic energy for selected body parts
- Plotting the angles or the kinetic energy of selected joints
- Selection with the mouse or sliders of a region of the data file to be plotted.
- Segmenting of either the velocity or kinetic energy for body parts or of joint angles or kinetic energy for joints.

To get data into the visualizer data is captured from the body model server and saved to a body_data format file; this file is converted to a Matlab matrix file by a conversion program.

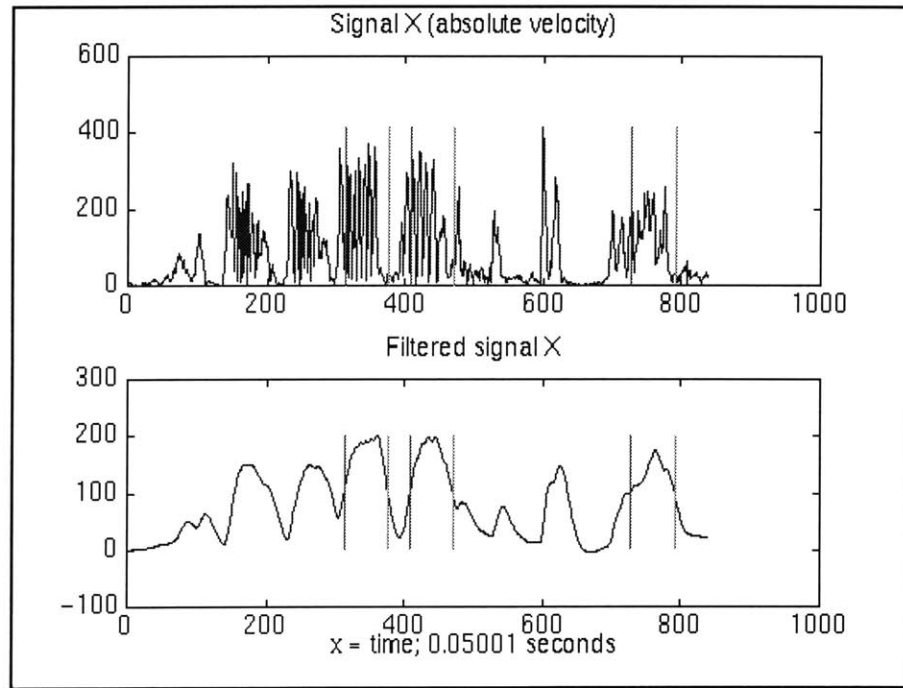


Figure 3.10 Velocity of wrist filtered at 1/3Hz and segmented at 100 cm/sec.

Figure 3.9 shows the interface to the analyzer. The user can select (with the push-buttons) various dynamical characteristics (velocity, joint angle velocities) of various body parts, and have these plotted as a function of time. The slider bars or mouse allow one to select part of the motion file to be looked at. The selected parts of the motion are then segmented into regions of action and no action. This is done by low-pass filtering with a chosen cutoff frequency followed by segmentation at a chosen threshold (see Figure 3.10).

3.4.2 Segmentation and selection of body motions

The segmentation technique works well with one signal but one finds many overlapping segments in the signals from the shoulders, elbows, wrists, and

fingers which all tend to move together. We would like to focus attention on the stream that presents the most visual salience. On the premise that large, fast moving objects will grab our visual attention we use mass and velocity to differentiate among the various moving body parts.

Three different models of attention were tried: momentum, kinetic energy, and power. We first used the momentum of the body part, or mv , to focus the segmentation. The more massive parts were always selected because the velocities of the different parts were comparable while their masses were very different.

To favor faster moving parts, kinetic energy, or $mv^2/2$, was tried. The kinetic energy biases appeared to be a useful gauge of perceptual salience when viewed with a graphical data viewer, DataView (see below). To determine the kinetic energy of a body part one employs the standard equation: $K = (1/2)mv^2$. At first each body part was assigned a mass value proportional to its size, e.g. finger = 1, hand = 10, head = 20. The velocity is determined from the displacement of a reference landmark. For body parts where position information is not known, joints angle velocity may be used instead, e.g. for the fingers. The kinetic energy for a one degree-of-freedom joint angle, θ , is calculated by, $K_j = \frac{1}{2}md^2\dot{\theta}^2$ where m = mass of distal segment and d = length of distal segment [38].

Finally the power of movement which measures the change in energy over time, was tried. However, the extra computation did not achieve much gain in perceptual accuracy over the kinetic energy model.

DataView, a body motion playback program, was used to visualize the above models of attention. It maps data files onto a graphical body and colors the

segments of the graphical body with intensities of red corresponding to the calculated momentum/energy/power values calculated for that body part.

A modified kinetic energy measure of joint angle motion was chosen as the method of segmentation in the BeeSystem because of its selection of visually salient body parts. The md^2 components used are 1, 0.75, 0.2 and 0.25 for the shoulder, elbow, wrist and finger joints respectively. These were arrived at after using the approximate real mass and length values (normalized to the length and mass of the upper arm) and then biasing until the segmentation matched visual inspection. The segmenter's cutoff power was 20 units.

4 The Working Prototype Systems

The first prototype system was built using the building block components discussed in chapter 3. Its purpose was to drive the Graphical Body (GBody) with the data from the Body Model Server (BMS). The result was real-time control of the GBody by the suited-user. This was accomplished through mapping functions that apply the joint angles from the BMS Client (BMC) to the update-joint-angle procedures of the GBody. Functions exist for direct (left = left), mirrored (left=right) and a puppet map (where the user's hand controls the movement of the whole figure). The facility with which joint angles could be re-mapped onto different body parts showed the power of joint angle representation of motion⁹.

The power of direct control becomes a liability, however, when we want to direct complex repetitive movements. The director must do everything. The second prototype system, designed with this in mind, enables the user to direct the flight of an animated bee. For example the director can tell the bee to fly and give an example wing-flap with his arms. The system responds with its normal flying behavior with the wing-flap of the user substituted for its own. The first prototype, the Direct Motion Mapping System is described next followed by Interpretative Motion Mapping System.

⁹Similar observations have been made by Sturman with respect to hand data[45]

4.1 Direct Motion Mapping System (DMMS)

The shaded elements in the lower part of Figure 4.1 have been described in Chapter 3. We focus here on the four central parts of the DMMS.

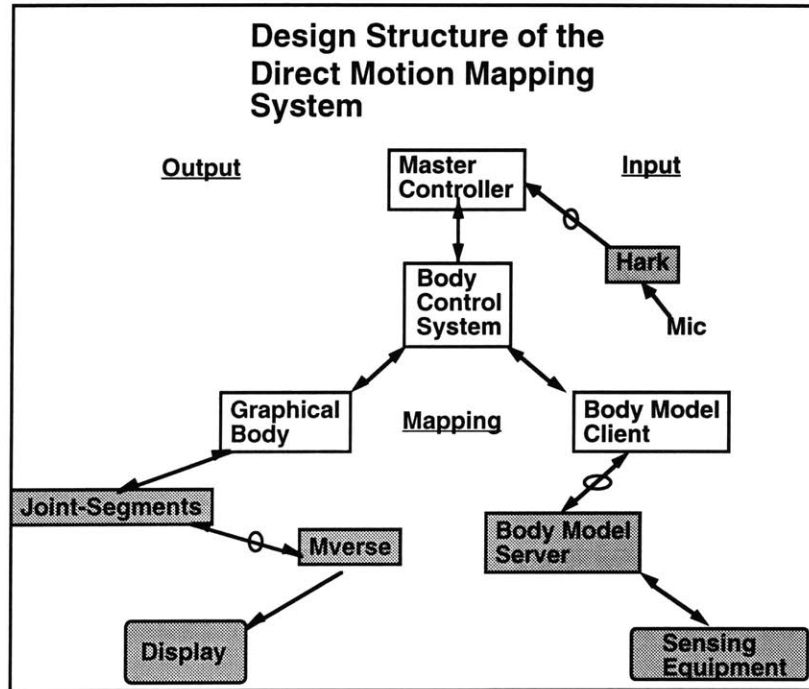


Figure 4.1: Links with circles around them separate asynchronous processes. These processes communicate through network sockets allowing them to reside on different machines.

- The Graphical Body (GBody) is a hierarchy of joint-segments that make up an upper torso, including: trunk, arms, hands, thumbs, head and eyes. Control of the body's posture is through the joint-segment's update-joint-angles procedures (see subsection 3.3.2).
- The Body Model Client (BMC) provides access to raw and normalized data from the BMS. There are three levels of access to the BMS data provided by the BMC: raw data level; derivative of data (velocities); and second derivative of data (acceleration). It has three modes of operation: asynchronous mode, a polled (synchronous) mode, and a data file mode. In the asynchronous mode the client specifies a data

rate for the server to send out data. In polled mode the server waits for the client to send it a request for data. In the last mode, data file mode, the BMC reads from a stored file. For all modes the client specifies through a bit-mask which regions of the body it wants data for.

- The Body Control System (BCS) maps motion from the Body Model Client to the Graphical Body. Motion is represented by the joint angles and position of the body from the body model server. These values are then used as parameters to the GBody's `update_joint_angle` functions. Several mapping options are available: direct, mirror, and puppet. These can be selected at any time to change the way motion is mapped from the BMC to the GBody.
- The Master Controller (MC) is a finite state machine that manages the state of the system. It handles speech commands from Hark and directs the Body Control System. Through speech the user may change the type of mapping, request a change in view-point, or request that motions be recorded or played back, or ask for a new graphical body. Multiple graphical bodies can appear in the same display by instantiating new BCS's. The additional BCS must get their data from a pre-recorded data file since only one connection at a time is allowed to the Body Model Server.

Table 4.1: Commands of the DMMS (selected through speech recognizer)

| Speech Command | Action |
|-----------------------|---|
| "map the body" | begins the real-time mapping of movements onto graphical body |
| "stop mapping" | stops the mapping. |
| "direct map" | map left = left, one-to-one |

| | |
|---|---|
| "mirror map" | map left = right |
| "puppet map" | map left hand onto GBody |
| "color motion map" | color segments according to their energy |
| "animate the view", "reset view", "recenter body" | controls the camera and placement of GBody. |
| "record motion" | save a motion to a file |
| "playback motion" | playback a saved motion on GBody |
| "create a partner" | make another GBody |
| "interact with partner" | control one GBody live and the other from a saved file |
| "direct a bee" | switch to the interpretative mapping system (see section 4.2) |

Figure 4.2 shows the author driving the Graphical Body through the DMMS. The second prototype, which we describe next, takes the idea of re-mapping motions a step further by mapping across articulations: Human to bee.

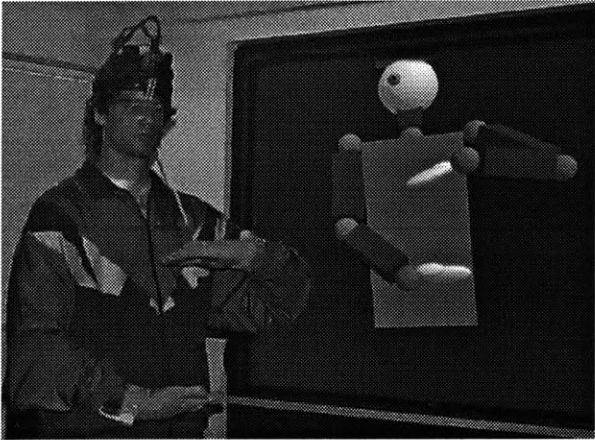


Figure 4.2: The author wearing the sensing equipment. In the background is a graphical model driven by the joint angle output of the BMS.

4.2 Interpretative motion mapping: The BeeSystem

The BeeSystem breaks the one-to-one real-time control of the DMMS. For this the DMMS is modified as follows: (a) the GBody is replaced by the Bee and its Motor Programs; (b) segmentation of data is inserted between the BMC and the MC; (c) the MC is modified to direct the Bee from the segmenter output and Hark output; and (d) the BCS, which performed the direct mapping, is no longer needed and hence removed. By adding knowledge of motion at both the input and output ends of the system can replace the direct-driving behavior with interpretative direction. Figure 4.3 illustrates the new architecture with the new elements unshaded.

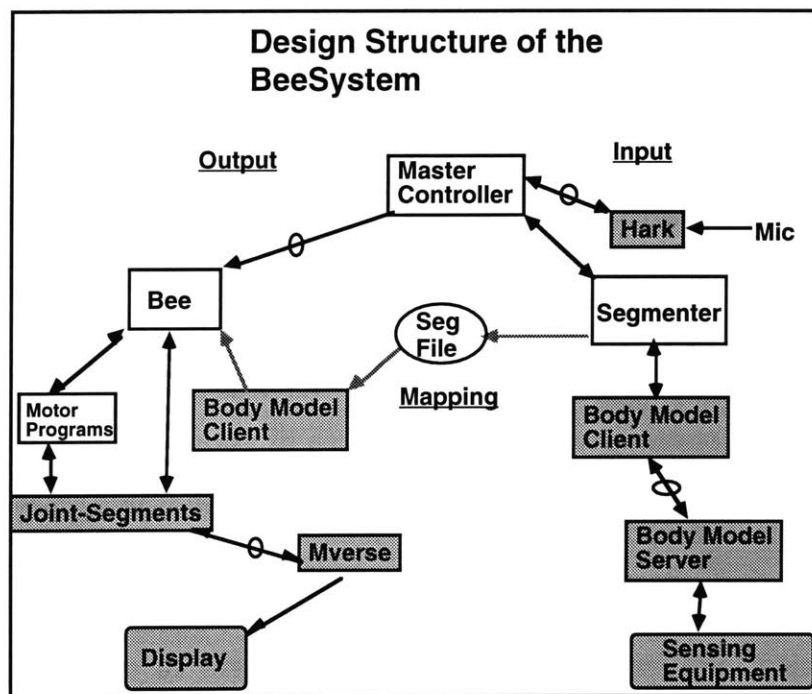


Figure 4.3: Architecture of second prototype system.

4.2.1 Segmentation of body data

The immediate question that arises is which features to look for in the input from the director. At first I thought that we should extract the frequency and

amplitude of periodic movements by the director. Then these features could be applied to the flight of the virtual bird. After some thought and consultation with potential users of such a system it was decided that these two features of control didn't empower the director anymore than having knobs or sliders in a mouse based interface. So instead I chose to extract the raw signal of the director's gestures and apply this to the virtual bee's motion.

A raw signal means data from either the position or the joint angles of the director's body parts. The choice depends on what one wants to control at the other end. For example to control the up and down path of the bee one would probably want to look for activity in the vertical coordinate of the position data from the director's body movements. However, to control the flapping of the wings of the bird it is more useful to look at the angle values from the joints of the arms of the director, which is what is actually done in the system. In the demo system wing flapping is guided by the director's example, so we segment the joint angles of the director's arm.

The scheme used for segmenting the director's gestures is the filter and threshold method described in section 3.4.2. The segmentation technique uses the kinetic energy of joint angle motion to select the most perceptually salient motion among the shoulder, elbow, wrist and fingers.

The first derivative of the shoulder elevation, elbow flex, wrist pitch, and finger flex angles for both arms are each squared, multiplied by the square of the length of its distal segment and added together, and multiplied by half the mass of each and then low-pass filtered with a 1/3 Hz cutoff frequency (see equation 3.1). This produces four streams of data representing the kinetic

energy produced through activity in left-right joint-pairs of the director's arms.

Thresholding of the filtered streams picks out the largest value and compares it to a predetermined minimum energy level. If the stream surpasses this level then the segment begins and data is recorded to a temporary disk file until the energy drops below the threshold level. If the length of a captured segment exceeds the filter cut-off period (three seconds) then it is moved to the seg file for reading by the Bee.

4.2.2 The bee and its motor programs

The Bee integrates two levels of control over an animated bee; joint-level and motor-program control. Clients of the Bee may request a predefined action by calling a motor-program or it may control a specific degree of freedom with data from a file.

Table 4.2: Commands provided by the Bee.

| Command Name | Parameters |
|---------------------|--|
| bounce | speed, height |
| control | map_to bee DOF, map_from body DOF |
| flap | left_range, left_freq, right_range, right_freq |
| fly | bounce_ht, freq, flap_range |
| hold | body_part, joint_angles |
| puff | vertical_scale, horiz_scale |
| turn | axis, degrees |
| shake | freq, up_down(?) |
| stop | |
| quit | |

Calls to motor programs must be accompanied by the appropriate parameters. Table 4.2 lists the Bee's motor programs and parameters. These parameterized motion control functions give the client high level control over the Bee's behavior (see Section 3.3.2).

Alternatively the client can control the Bee's posture directly at the joint angle level by using the "control" command. The parameters to the "control" function are the bee joint to control, e.g., wing, and which joint from a body data file to use as the controller, e.g. shoulder. Each cycle of the animation the data file is read for the new angle values, and these values are normalized and then mapped onto the `update_joint_angles` procedures of the appropriate joint-segments of the Bee.

All bee commands take the repeat parameter which says how many times to cycle through the motor program or through the data file for control commands.

4.2.3 Mapping from input to output

The MC maps the output of the segmenter and the output from hark into appropriate calls to the Bee. For example if the director says "fly" the system will send a fly command to the Bee, however, if the user says "fly like me" it will wait for the segmenter to return a gesture. When a gesture is found the MC will make a control request to the Bee specifying the wings and the joint angle returned by the segmenter.

For example:

User: "Fly."

System: sends a "fly 30 0.5 0.5 0.5 0.5" to the Bee. The Bee sends the appropriate motor commands to the Mverse system causing the bee to fly around performing 30 cycles of wing flapping and up and down motion.

User: "Fly like this [flapping motion with wrists (3 sec.)]."

System: sends a "bounce 30 0.5 0.5 0.5 0.5" and a "control n wings wrist" to the Bee. The Bee then sends a bounce motor command to Mverse and synchronizes it with the playback of the stored gestural example which is mapped directly onto the wings of the bee. The playback is repeated n times, proportional to the length of the captured motion segment.

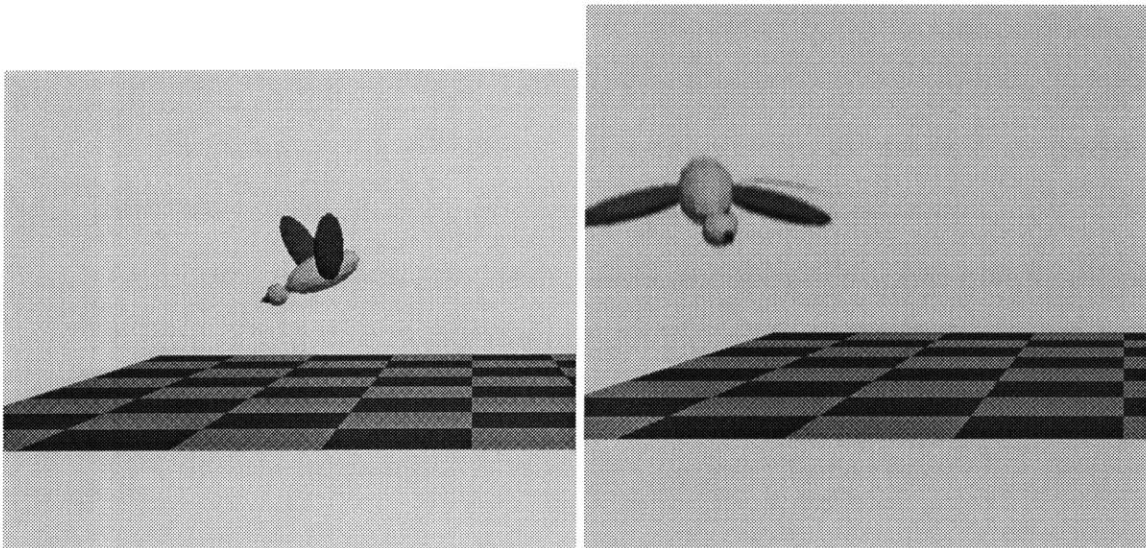


Figure 4.4: The virtual bee in flight.

4.3 Summary

In this chapter, I have presented two working prototype systems: the direct body motion mapping system (DMMS) and the interpretative motion mapping system (BeeSystem). The DMMS enables a user to directly manipulate a graphical representation of a human upper body through his body movements. It affords multiple mappings (direct, mirrored, puppet) and allows motions to be recorded and played back using different mappings. The

BeeSystem enables a user to direct the flight of a bee through gestural example. Speech without gesture causes a default flying behavior, whereas in combination the gestural example is extracted through a segmentation process and then applied to the wings joints of the graphical bee while motor programs control the other parts of flight, forward and up and down motion.

A follow-up work to this thesis should test this prototype with real users and compare the ease of use of this prototype system with a keyboard/mouse based interface. A timed task completion exercise and subjective survey could be used as comparison measures.

5 The Next Step

Obviously the prototypes described in sections 4.1 and 4.2 are limited in their functionality. Here we consider some of the limitations in the present design:

- The tight coupling of the speech parsing with the gesture segmentation has its tradeoffs. A close connection is good because the command may help constrain the location or type of gesture to expect. The drawback is that frequently these two streams of data will need simultaneous attention. Much as the ears and the eyes operate concurrently in people, the two streams should be handled asynchronously by the system. In the current system the cycle rate is fast enough to prevent interference between gesture segmentation and speech parsing. However, if either part were more complex the interaction of the delays would cause interference problems. In general the centralized structure of the system limits its power. Although the system is broken up into functional units, they cannot operate in parallel as they rely on the master controller (MC) to call them. In a decentralized (asynchronous) system the inputs would be handled by separate processes.
- The speech parsing capabilities of the system are severely limited. Modifiers (i.e., "faster", "higher", etc.), which are logical extensions to the command vocabulary, are not possible because there is no history of commands maintained in the finite state machine of the MC.

- The finite state machine parsing design in the MC does not scale up as extensions to the grammar allowed dramatically increases the number of states needed.
- Static descriptions of body posture are not possible with a system that only segments periods of motion.
- Synchronization of the playback of the director's example with the execution of the Bee's motor programs is not possible. Each has its own period.

Possible solutions to these problems are presented in the next section. It outlines an architecture that would extend the functionality of the current system. The intent is to thoroughly outline the design so that someone might construct a new enhanced prototype system.

This section details the next step envisioned for the work of this thesis. The second prototype (Interpretative Motion) system points in the right direction but has some limitations. Specifically, the next step should attempt to enable multiple dynamic commands, permit static postural control of the bee, allow speech modifiers, i.e. adverbs to modify motions, and synchronize the director's examples with motor behaviors in the Bee. A modified architecture is presented which would enable these additions to the functionality of the present system.

5.1 Multiple dynamic commands

In order to make all of the Bee's motor programs accessible to the director three modifications must be done. (a) Extend Hark's vocabulary of recognized word to include the new commands. (b) Augment the segmentation scheme

to look for new features. This is done by adding new degrees-of-freedom to its filter and threshold scheme (see 4.2.3). Specifically, to enable the "bounce" command the segmenter should look for vertical (z-coordinate) activity in all body parts.

5.2 Static posture control of Bee

Permitting postural commands such as, "hold the head like this", requires three changes to the present system: (a) Additions to the speech vocabulary and parsing must be made to allow for body part specification; (b) Static poses need to be represented within the segmenter; (c) The addition of new functions to the MC that map from static poses to "hold" commands for the Bee.

5.3 Allow speech modifiers

To permit the qualification of commands, e.g. "fly quickly", or just "slower", requires three changes to the system: (a) Add to the MC parser a memory for the last commands sent to the Bee; (b) Enhancement of Hark vocabulary and parsing to include modifiers; (c) Addition of functions to the MC that perform the modifications to the parameters to the Bee commands.

5.4 Synchronization of motor and director control

Synchronized mapping of the director's example gesture with the motor behaviors of the Bee requires one of two changes. One could extract the parameters of motor program from the segmented example through featural analysis. For example if the command was "bounce" the amplitude and frequency of the director's up and down gesture could be extracted, normalized and used as the displacement and frequency parameters to the bounce motor program. Alternatively, the motor program's period could be

scaled to match that of the director's example by extracting its period (using autocorellation).

5.5 Conclusion

This thesis has contributed to three different areas:

1. Body sensing and representation for real-time interaction: the Body Model Server
2. Segmentation and selection of body movements: kinetic energy model of visual salience.
3. Multi-modal interfaces: the BeeSystem explores director-actor style communication through the integration of high and low levels of control over the animation of an articulated graphical creature.

Appendix A. Director-actor Pilot Experiment

I conducted the following experiment to examine the use of gesture and speech between human directors and actors. The setup can be seen in Figures A.1 and A.2.

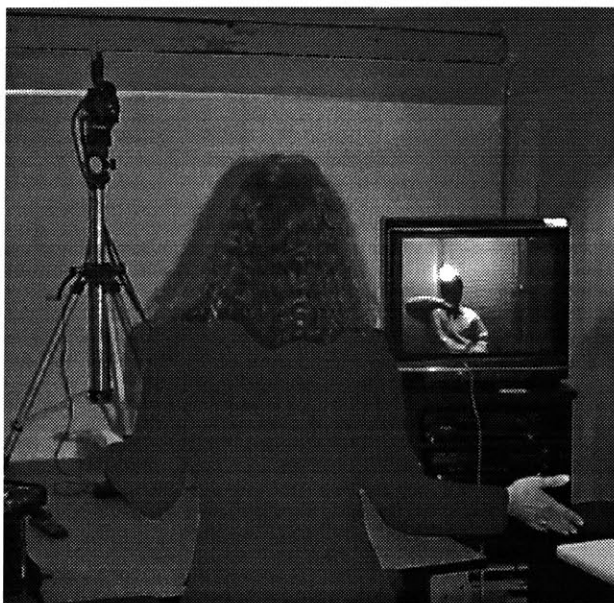


Figure A.1: The actor's room. The camera is in the upper left corner, the monitor is in the middle right showing the director.

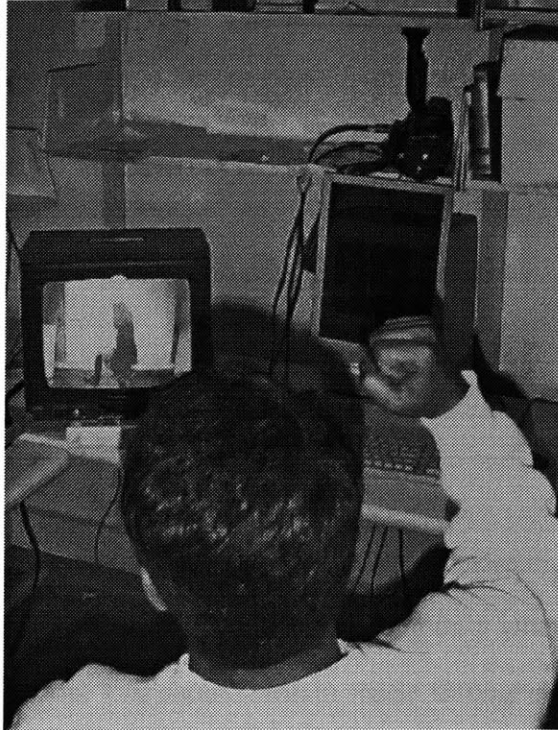


Figure A.2: The director's room. The camera is in the upper right, the monitor is in the middle left.

A.1 Experimental Protocol

A.1.1 Experimental Setup:

A video camera and a television monitor was setup in each of two adjoining rooms so that the output of the video camera in one room was connected with the input of the TV in the adjacent room. One room was designated the director's room and the other the actor's. In this way I was able to simultaneously record the actions of both the actor and the director.

A.1.2 Procedure Description:

An actor and director are paired. The actor goes into the actor room and watches the video monitor. Before each session the director watches a demonstration video showing a person performing an example walk. He then sits on a stool in front of a

video camera and directs the actor in the other room to walk in the manner just seen. The actor can both see and hear the director, and vice-versa, by way of the monitor-camera hookup. The only constraint is that the field of view is limited by the camera lens. The director must also remain seated throughout the session. Each actor/director pair will go through a few of these sessions.

A.1.3 Instructions:

To the director:

"Watch the demonstration of the walk on the video monitor. Then direct the actor to move like the person in the video. Stop directing only when the actor's movements meet your satisfaction. You should remain seated throughout the experiment."

To the actor:

"The director will try to get you to move in a particular way: follow his/her directions to the best of your ability. Use the space in the room to demonstrate the walking."

After reading the instructions the subjects were asked to read and sign an informed consent form that follows:

CONSENT DECLARATION

I fully understand that my participation in the experiment titled "Directing Walking Behavior through Gesture and Speech" is voluntary and that I am free to withdraw my consent and to discontinue participation at any time without prejudice to myself. The experimental procedures and their purposes have been explained to me and the Investigator (Joshua Bers) has offered to answer any inquiries concerning the procedures.

In the unlikely event of physical injury resulting from participation in this research, I understand that medical treatment will be available from the M.I.T. Medical Department, including first aid emergency treatment and follow-up care as needed, and that my insurance carrier may be billed for the cost of such treatment. However, no compensation can be provided for medical care apart from the foregoing. I further understand that making such medical treatment available; or providing it, does not imply that such injury is the Investigator's fault. I also understand that by my participation in this study I am not waiving any of my legal rights.

I understand that I may also contact the Chairman of the Committee on the Use of Humans as Experimental Subjects, M.I.T. 253-6787, if I feel I have been treated unfairly as a subject.

Consent to be Videotaped

Your actions performed during the experiment will be recorded so that the investigator (Joshua Bers) can analyze them at a later time. Viewing of the raw footage of these tapes will be limited to the investigator. Any dissemination of the footage will be edited to insure the privacy of the subjects.

I have read and understand the above and agree to participate in this research effort. I consent to be videotaped as part of this study.

Name of Subject

Signature Date

A.2 Results and observations of experiment

| Speech | Type of Gesture | Viewpoint of Gesture |
|----------------------------|------------------------|-----------------------------|
| walk | Directionality | Observer |
| arms like this | Manner of Motion | Observer |
| fold arms | Manner of Motion | Observer |
| more fold | Manner of Motion | Character |
| feet come up | Manner of Motion | Observer |
| little bit more of raising | Manner of Motion | Observer |
| raise your arms | Manner of Motion | Character |
| 0 | Manner of Motion | Character |
| 0 | Manner of Motion | Observer |
| raise your knees | Manner of Motion | Observer |

| | | |
|---|------------------------|-----------|
| raise your arms | Manner of Motion | Character |
| you were doing | Manner of Motion | Character |
| he was doing like that | Manner of Motion | Character |
| he had his hands open | Manner of Motion | Character |
| lift your knees a little more | | 0 |
| shake | Manner of Motion | Character |
| stand on your toes | Manner of Motion | Observer |
| swim with your arms | Manner of Motion | Character |
| play with your hips | Manner of Motion | Character |
| lift your knee and extend like | Manner of Motion | Observer |
| kick a little bit higher now | Manner of Motion | Observer |
| swinging your arms from the elbow | Body Part Concerned | Character |
| step a little bit more | Manner of Motion | Observer |
| jump across | Manner of Motion | Observer |
| jump up and split your legs | Manner of Motion | Observer |
| jump land | Manner of Motion | Character |
| somersault across | Manner of Motion | Character |
| swing both arms and legs | Manner of Motion | Character |
| balancing yourself | Manner of Motion | Character |

| | | |
|---|---------------------|-----------|
| putting your toes on one and then another | Manner of Motion | Observer |
| slow it down | Manner of Motion | Character |
| you're swaggering | Manner of Motion | Character |
| winding back and forth | Directionality | Observer |
| you're head is down | Body Part Concerned | Character |
| zig-zag | Directionality | Observer |
| smoother | Manner of Motion | Character |
| turn while you walk | Manner of Motion | Character |
| glide across | Directionality | Observer |
| kick forward and step | Manner of Motion | Observer |
| legs go higher | Manner of Motion | Observer |
| bend | Manner of Motion | Observer |

Frequencies (out of 40):

Viewpoint: Character 20; Observer 20

Manner of Motion 34; Body Part Concerned 2; Directionality 4

Table A.1: Iconic gestures that co-occurred with speech from the pilot experiment.

This was informal study. The selection of subjects was not random, and small in number (four). Two sessions were run (one for each director-actor pair) each consisting of five example walks. Table A.1 shows data on iconic gestures that co-occurred with speech.

Appendix B. Data Format for BMS Clients

```
// This module defines the body model interface
```

```
// Defines for expressing interest in possible data values
```

```
// the bits of an unsigned long
```

```
#define LA 0x00000001 // left arm (shoulder->wrist)
```

```
#define LH 0x00000002 // left hand (palm->fingers)
```

```
#define RA 0x00000004 // right arm
```

```
#define RH 0x00000008 // right hand
```

```
#define EYE 0x00000010 // eye
```

```
#define HED 0x00000020 // head
```

```
#define TOR 0x00000040 // torso
```

All data values are 2 byte-ints biased by 1.0e1
except unit-vector values which are biased by 1.0e4

```
/* In general the order of data for a particular body part is:
```

```
  x, y, z position in the space of the transmitter cube
```

```
    X is toward the screen
```

```
    Y is to the left as you face the screen
```

```
    Z is up.
```

Angle data follows for joints (number of values depends on degrees of
freedom for the particular joint),

followed by unit orientation vectors for the segments: Normal(x,y,z)
and Approach(x,y,z).

Normal points perpendicular to the body segment. Approach points along
the main axis of the body segment.

```
*/
```

```
/* Data bytes in order:
```

```
  Arm data is shoulder, elbow, wrist
```

```
Shoulder:
```

```
  X, Y, Z (of shoulder point),
```

Elevation, Abduction, Twist;

Normal x,y,z and Approach x,y,z of upper arm.

Elbow:

X, Y, Z,

Flex, Twist;

Normal x,y,z and Approach x,y,z of forearm.

Wrist is

X, Y, Z,

Pitch, Yaw

Joint angle descriptions:

All angles in degrees

Shoulder

Elevation 0 with arms relaxed at side, up is +

Abduction 0 with upper arm perpendicular to shoulder, away from body is +, across body is -

Twist 0 with arms hanging at sides, inside of elbow facing forward, inside of elbow away from body is +

Elbow

Flex 0 when arm straight, wrist toward shoulder is +

Twist 0 when wrist axis is 90 degrees from elbow axis, thumb out is +, thumb in is -

Wrist

Pitch 0 when back of hand level with forearm, hand up is +, hand, down is -

Yaw 0 when palm aligned with forearm, fingers toward thumb is -, away from thumb is +

Hand data is palm, thumb, fore, middle, ring, pinkie fingers

Palm is X, Y, Z, Normal, Approach, palm arch

Normal points perpendicular to the palm

Approach is the vector pointing down the hand towards the fingers

Palm arch 0 with palm flat on table, closed is +

For all finger joints the range is 0-90 degrees

except for the abduction values which range from 0-30

Thumb is mcp, ip

mcp 0 with thumb 90 degrees from palm

ip 0 with thumb extended

Forefinger is mcp, pip

mcp 0 with finger extended

pip 0 with finger extended

Middle finger is mcp, pip

mcp 0 with finger extended

pip 0 with finger extended

Ring finger is mcp, pip

mcp 0 with finger extended

pip 0 with finger extended

Pinkie finger is mcp, pip

mcp 0 with finger extended

pip 0 with finger extended

Abduction between fingers

Thumb and fore

Fore and middle

Middle and ring

Ring and pinkie

Eye is X, Y, Z, elevation and azimuth

X, Y, Z are in global coordinates, give center of eye

Elevation is 0 looking flat ahead + up

Azimuth is 0 straight in front + to the left.

Head is X, Y, Z, elevation, abduction, twist, Normal, Approach.

X, Y, Z are in global coordinates, give top of crown

Elevation 0 with spine straight, up/back is +

Abduction 0 with head in line with axis of spine, right is +

Twist 0 with nose in line with bellybutton, right is +

Normal x, y, z give the direction unit vector for the face

Approach x, y, z give the head up.

Torso is X, Y, Z, roll, pitch, yaw, Normal x, y, z, Approach x,y,z

```
    X, Y, Z are in global coordinates, give center of spine at mid-  
    torso  
roll, pitch, yaw are the euler angles.  
Normal x, y, z points forward  
Approach x,y,z points up the spine.  
*/  
typedef unsigned short BmData;      // The BodyModel Data type
```

Appendix C. Joint Angle Calculations for the Body Model

C.1 Hand and wrist

The cybergloves measure the following degrees of freedom for the hand and wrist: two flexion angles for each finger and thumb, four abduction angles, one between adjacent fingers, as well as the wrist angles, pitch and yaw. During calibration we determine the offset and gains that linearly map each sensor output onto a pre-determined angular range. These values are applied to the output of the corresponding sensors using Equation C1 to get joint angles.

Equation C1: $\text{angle} = (\text{sensor value} - \text{offset}) * \text{gain}$

where

$\text{gain} = (\text{angular range}) / (\text{sensor range})$ and

$\text{offset} = \text{min. sensor value} - (\text{min. angle}) / \text{gain}.$

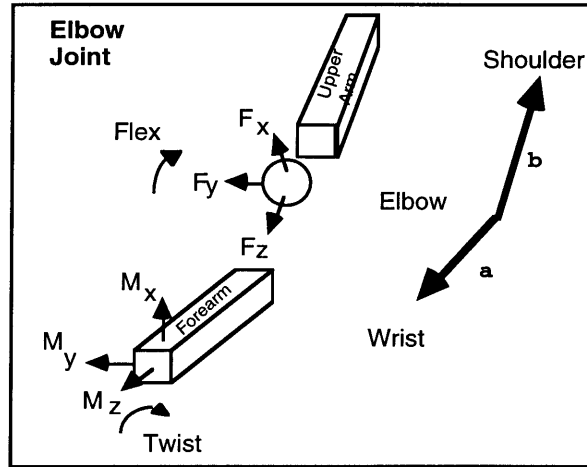


Figure C.1: Expanded view of elbow joint showing the fixed and moving coordinate frames. On the right is a vector representation.

C.2 Elbow

Figure C.1 shows an expanded view of the elbow joint and its associated segments with coordinate frames. First, we use transform equations described in section 4.2.4 to find vectors from the elbow to the wrist and to the shoulder, and then we calculate the angle between them to get the flex of the joint using equation C2.

Equation C2: Elbow flex = arcsine($\mathbf{a} \cdot \mathbf{b}$)

In equation C2, the unit vectors \mathbf{a} and \mathbf{b} give the direction of the wrist and shoulder from the elbow, and \cdot specifies the dot-product vector operation. Elbow twist actually occurs between the wrist and the elbow in the radius and ulna bones, but for simplicity it is modeled as a DOF of the elbow joint. Twist is calculated by finding the angle between the wrist normal, M_x in Figure C.1, and the normal to the forearm in the plane formed by the forearm and upper arm. This normal is found by taking the cross-product of the forearm vector and the upper arm vector and crossing the result (F_y) with the forearm vector. Equation C3 shows the computation of these angles.

Equation C3: Elbow twist = angle between **wn** and **en**

wn = wrist normal, pointing away from palm. Obtained from sensor on the back of the wrist.

en = elbow normal pointing away from body = $(\mathbf{a} \times \mathbf{b}) \times \mathbf{a}$.

C.3 Shoulder

Figure C.2 gives an expanded view of the shoulder joint. The shoulder has three degrees of freedom: elevation, abduction and twist. The elevation is the unsigned angle between the upper arm vector and the z-axis of the shoulder joint, which runs straight down in line with the torso. Abduction is the signed angle between the projection of the upper arm vector onto the x-y plane of the shoulder joint and the x-axis. Twist is the difference in axial rotation about the z-axis of the upper arm that compensates for the ambiguous elevation, abduction pair [46].

Twist is calculated by rotating the shoulder reference frame by the measured elevation and abduction and then comparing that frame with the coordinate frame of the upper arm calculated from the wrist, using the transform equation technique described at the beginning of this section. The difference between the two transforms is a rotation about the z-axis (major) of the upper arm. As a result the twist angle is dependent on the values of both elevation and abduction; swinging one's elbow back and forth as it hangs straight down below one's shoulder causes the twist to jump from zero to 180 degrees. This discontinuity of the twist angle can be removed if we break the twist into its constituent parts [46].

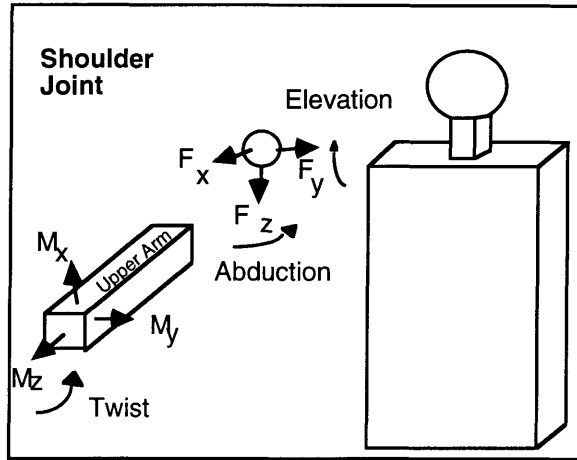


Figure C.2: The shoulder joint expanded, showing its three degrees of freedom.

A technique similar to shoulder angle calculation is used to obtain the elevation, abduction and twist angles of the neck.

C.4 Eye

The data from the eye tracker is fed into an interpolation routine to compute the elevation and azimuth angles. The interpolation function uses a nine point calibration table created by the user at system start-up. From this table, we construct two tilings for the eye data, one for elevation and one for azimuth. The tilings divide the data into eight triangular regions of linear interpolation. For each new data point (x,y) we find the tile that it falls under and use the equation for the tile's plane as the interpolation function. See [42] for details on this technique.

References

1. Holloway, R., *Z is for Zagreb*. 1972, London: Tantivy Press.
2. Church, R.B., et al. *The Development of the Role of Speech and Gesture in Story Narration*. in *biennial meeting of the Society for Research in Child Development*. 1989. Kansas City, MO.
3. McNeill, D., *Hand and Mind: What Gestures Reveal about Thought*. 1992, Chicago: University of Chicago Press.
4. Cassell, J. and D. McNeill, *Gesture and the Poetics of Prose*. *Poetics Today*, 1991. **12**(3): p. 375-403.
5. Rimé, B. and L. Sciaratura, *Gesture and Speech*, in *Fundamentals of Nonverbal Behavior*, R.S.F.B. Rimé, Editor. 1991, Cambridge University Press: New York. p. 239-281.
6. McNeill, D. and E. Levy, *Conceptual Representations in Language Activity and Gesture*, in *Speech, Place, and Action*, R. Jarvella and W. Klein, Editors. 1982, John Wiley and Sons: Chichester.
7. Cassell, J., *Comments on video tapes of experiment*, . 1995.
8. Whittaker, S. and M.A. Walker. *Toward a theory of multi-modal interaction*. in *AAAI Workshop on MultiModal Interaction*. 1991.
9. Wahlster, W., *User and Discourse Models for MultiModal Communication*, in *Intelligent User Interfaces*, J.W. Sullivan and S.W. Tyler, Editors. 1991, Addison-Wesley: New York, NY. p. 45-67.

10. Kurtenbach, G. and E.A. Hulteen, *Gestures in Human-Computer Communication*, in *The Art of Human-Computer Interface Design*, B. Laurel, Editor. 1990, Addison-Wesley: Reading, MA. p. 309-317.
11. Thompson, C., *Building Menu-Based Natural Language Interfaces*. Texas Engineering Journal, 1986. 3: p. 140-150.
12. Neal, J.G. and S.C. Shapiro, *Intelligent Multi-Media Interface Technology*, in *Intelligent User Interfaces*, J.W. Sullivan and S.W. Tyler, Editors. 1991, Addison-Wesley: New York, NY. p. 11-43.
13. Carbonell, J.R., *Mixed-Initiative Man-Computer Dialogues*, . 1970, Bolt, Beranek and Newman.
14. Koons, D.B., C.J. Sparrell, and K.R. Thórisson, *Integrating Simultaneous Input from Speech, Gaze and Hand Gestures*, in *Intelligent Multi-Media Interfaces*, M.T. Maybury, Editor. 1994, AAAI/MIT Press: Cambridge.
15. Bolt, R.A., *Put-That-There: Voice and Gesture at the Graphics Interface*. Computer Graphics, 1980. 14(3): p. 262-270.
16. Schmandt, C. and E.A. Hulteen. *The Intelligent Voice-Interactive Interface*. in *Human Factors in Computing Systems*. 1982.
17. Sparrell, C.J., *Coverbal Iconic Gesture in Human-Computer Interaction*, Master's Thesis. 1993, MIT.
18. Zeltzer, D., *Task-level Graphical Simulation: Abstraction, Representation, and Control*, in *Making Them Move: Mechanics, Control,*

- and Animation of Articulated Figures*, N.I. Badler, Barsky, B. A., Zeltzer, D., Editor. 1991, Morgan-Kaufmann: San Mateo, CA. p. 3-33.
19. Maxwell, D.R., *Graphical Marionette: A Modern-Day Pinocchio*, . 1983, MIT.
 20. Ginsberg, C.M., *Human Body Motion as Input to an Animated Graphical Display*, Master's Thesis. 1983, MIT.
 21. Bolt, R.A., *Funding Proposal for the development of a Graphical Marionette*, . 1981, MIT.
 22. Culhane, S., *Animation From Script To Screen*. 1988, New York: St. Martin's Press.
 23. Badler, N.I., C.B. Phillips, and B.L. Webber, *Simulating Humans: Computer Graphics Animation and Control*. 1993, New York, NY: Oxford University Press.
 24. Pentland, A., *et al. Visually Guided Animation*. in *Computer Animation '94*. 1994. Geneva, Switzerland.
 25. Blumberg, B.M. and T.A. Galyean. *Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments*. in *submitted to Siggraph '95*. 1995.
 26. Korein, J.U., *A Geometric Investigation of Reach*. ACM distinguished dissertations. 1985, Cambridge: MIT Press.
 27. Zeltzer, D., *Motor Control Techniques for Figure Animation*. *Computer Graphics and Applications*, 1982(October).

28. Chen, D.T., et al. *The Virtual Sailor: An implementation of interactive human body modelling*. in *Virtual Reality Annual International Symposium*. 1993. Seattle, WA: IEEE.
29. Essa, I., S. Sclaroff, and A. Pentland, *Physically-based Modelling for Graphics and Vision*, in *Directions in Geometric Computing*, R. Martin, Editor. 1993, Information Geometers: U.K.
30. Badler, N.I., M.J. Hollick, and J.P. Granieri, *Real-Time Control of a Virtual Human Using Minimal Sensors*. *Presence*, 1993. 2(1): p. 82-86.
31. Schiphorst, T., S. Mah, and J. Crawford. *StillDancing: Interacting Inside the Dance*. in *CHI '94*. 1994. Boston: ACM Press.
32. Van Cott, H. and R. Kinkade, *Human Engineering Guide to Equipment Design*, . 1972, US Government.
33. Fitts, P., *The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement*. *Journal of Experimental Psychology*, 1954. 47(June).
34. Brooks, T.L., *Telerobot Response Requirements*, . 1990, STX Corporation.
35. Card, S., T. Moran, and A. Newell, *The Psychology of human-computer interaction*. 1983, Hillsdale, NJ: Lawrence Erlbaum Associates.
36. Yarbus, A.L., *Eye Movements and Vision*. 1967, New York: Plenum Press.

37. Foley, J.D., et al., *Computer Graphics: Principles and Practice*. 2nd ed. Systems Programming Series, ed. I.E. Board. 1991, Reading, MA: Addison-Wesley.
38. Paul, R.P., *Robot Manipulators: Mathematics, Programming and Control*. Artificial Intelligence, ed. P.H. Winston and M. Brady. 1981, Cambridge, MA: MIT Press.
39. Denavit, J. and R.S. Hartenberg, *A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices*. Journal of Applied Mechanics, 1955. 77(June): p. 215-221.
40. Simms, K., *Locomotion of Jointed Figures over Complex Terrain*, Master's Thesis. 1987, MIT.
41. Different, N., A.R. Tilley, and J.C. Bardagjy, *Humanscale 1/2/3*, ed. H.D. Associates. 1974, Cambridge, MA: MIT Press.
42. Koons, D.B. and K.R. Thórisson. *Estimating Direction of Gaze in a Multi-Modal Context*. in 3CYBERCONF. 1993. Austin, TX.
43. Sanders, M.S. and E.J. McCormick, *Human Factors in Engineering and Design*. 1987, New York: McGraw-Hill.
44. Houy, D.R. *Range of Joint Motion in College Males*. in Conference of the Human Factors Society. 1983. Santa Monica, CA: Human Factors Society.
45. Sturman, D., *Whole Hand Input*, PhD Thesis. 1992, MIT.
46. Badler, N.I., J. O'Rourke, and B. Kaufman, *Special Problems in Human Movement Simulation*. Computer Graphics, 1980. 14(3): p. 189-197.