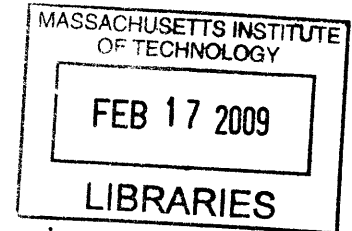


Affordable Avatar Control System for Personal Robots

by
Jun Ki Lee




Bachelor of Science in Computer Science and Engineering
Seoul National University, 2006

Submitted to the
Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements of the degree of
Master of Science in Media Arts and Sciences
at the Massachusetts Institute of Technology

February 2009

© 2008 Massachusetts Institute of Technology. All rights reserved.


Signature of Author


Program in Media Arts and Sciences
September 30, 2008

Certified by


Cynthia L. Breazeal
LG Group Career Development Associate Professor
Program in Media Arts and Sciences, MIT

Accepted by


Deb Roy
Chairperson
Departmental Committee on Graduate Studies

.

Affordable Avatar Control System for Personal Robots

by
Jun Ki Lee

Bachelor of Science in Computer Science and Engineering
Seoul National University, 2006

Submitted to the
Program in Media Arts and Sciences,
School of Architecture and Planning, on September 30, 2008
in partial fulfillment of the requirements of the degree of
Master of Science in Media Arts and Sciences
at the Massachusetts Institute of Technology

February 2009

ABSTRACT

Social robots (personal robots) emphasize individualized social interaction and communication with people. To maximize communication capacity of a personal robot, designers make it more anthropomorphic (or zoomorphic), and people tend to interact more naturally with such robots. However, adapting anthropomorphism (or zoomorphism) in social robots makes morphology of a robot more complex; thus, it becomes harder to control robots with existing interfaces. The Huggable is a robotic Teddy bear platform developed by the Personal Robots Group at the MIT Media Lab. It has its specific purpose in healthcare, elderly care, education, and family communication. It is important that a user can successfully convey the meaningful context in a dialogue via the robot's puppeteering interface. I investigate relevant technologies to develop a robotic puppetry system for a zoomorphic personal robot and develop three different puppeteering interfaces to control the robot: the website interface, wearable interface, and sympathetic interface. The wearable interface was examined through a performance test and the web interface was examined through a user study.

Thesis supervisor: Cynthia Breazeal

Title: LG Group Career Development Professor of Media Arts and Sciences

Affordable Avatar Control System for Personal Robots

by
Jun Ki Lee

Advisor

Dr. Cynthia L. Breazeal
LG Group Career Development Professor
Associate Professor of Media Arts and Sciences
MIT Media Laboratory

Reader

Dr. Rosalind W. Picard
Co-Director of Things That Think Consortium
Professor of Media Arts and Sciences
MIT Media Laboratory

Reader

Dr. Joseph Paradiso
Sony Corporation Career Development Professor of Media Arts and Sciences
Associate Professor of Media Arts and Sciences
MIT Media Laboratory

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Cynthia Breazeal. She gave me the chance to study this wonderful field in Academia, Human Robot Interaction (HRI). I borrowed many insights from her in HRI when I was developing many platforms in this thesis.

I also would like to thank my two other thesis readers: Roz Picard and Joe Paradiso. Roz gave me invaluable advices all the time when I visited her. They helped me both personally and academically. Joe's insights in Sensor Technologies greatly helped me developing both the wearable technology and the sympathetic interface.

I would like to thank Dan Stiehl who provided me this wonderful robot, the Huggable. I learned a lot from him including his endless devotion to work and many insights in our field of study.

I cannot help but thank the other Team Huggable members: Robert Toscano, Heather Knight, Allan Maymin, Jessica Hamrick, Kristopher Dos Santos, and many other UROPs who helped us creating one of the world's most sophisticated robot creatures. I also want to appreciate Daniel Bernhardt for his devotion to develop the part of the wearable interface.

Additionally, I thank Jesse in helping me find the ways to explore the sea of the C5M code base. I thank Phillip for the minutes that he spent to hear all my complaints. I also would like to thank the rest of Personal Robots Group's current and former members: Mikey Siegel, Sigurður Örn, Angela Chang, Matt Berlin, Andrea Thomaz, Guy Hoffman. Lastly, I thank Polly Guggenheim for her support to our group.

I also appreciate the support from the Samsung Scholarship. They provided me two years of scholarship to finish my Master's study.

After all, I became a husband and a father while I was writing my thesis. It was a big change and also a challenge for me to write my first thesis and to become a husband and father. Hyewon Kim, my wife, spent a tremendous amount of time together to overcome many hard days. Without her, I would not be able to accomplish anything. I also thank my parents and my sister who always gave me the endless support.

TABLE OF CONTENTS

1. INTRODUCTION	11
2. BACKGROUND.....	15
2.1. SOCIABLE ROBOTS.....	15
2.2. INTERFACES FOR TELE-EXISTENCE	17
2.3. HUMAN MOTION CAPTURE AND ANALYSIS.....	19
2.4. EMBODIED CONVERSATIONAL AGENTS	22
2.5. COMPUTER ANIMATION AND COMPUTER PUPPETRY	23
2.5.1 <i>Computer Puppetry</i>	23
2.5.2 <i>Motion Graphs</i>	25
2.6. TANGIBLE INTERFACES TO CONTROL VIRTUAL AND PHYSICAL AVATARS	27
3. THE HUGGABLE.....	29
3.1. HARDWARE SPECIFICATION.....	30
3.2. AUTONOMOUS MODE VS SEMI-AUTONOMOUS MODE	30
3.3. HUGGABLE SEMI-AUTONOMOUS CONTROL SYSTEM LAYOUT	32
3.3. MSRS SERVICES	33
3.4. C5M BEHAVIOR SYSTEM	36
4. WEB INTERFACE.....	41
4.1. SITUATION AWARENESS.....	41
4.2. FIRING CONTROL COMMANDS AND AUDIO INTERFACE.....	43
5. WEARABLE INTERFACE	46
5.1. PROPOSED APPROACH.....	46
5.2. RECOGNITION USING ACTIVE SENSING TECHNOLOGY	47
5.3. TRACKING AND POSE ESTIMATION	50
5.4. EVALUATION OF EARLY PROTOTYPES.....	55
6. SYMPATHETIC INTERFACE.....	62
6.1. HARDWARE	63
6.2. FEATURES.....	64
6.3. SOFTWARE SPECIFICATION	65
7. USER STUDY	67
7.1. STUDY DESIGN	67
7.2. DIALOGUE	68
7.3. QUESTIONNAIRE	70
7.4. RESULT AND USER RESPONSE	71
8. CONCLUSION	74
REFERENCES	76

1. Introduction

Unlike the past when robots were only seen on assembly lines in factories, we can now see personal robots everywhere. From Sony's AIBO to Lego Mind-storm kits, we currently interact with them as our companions, collaborators, housekeepers, pets and toys. As technology advances, the morphology of current personal robots is becoming more sophisticated, and the numbers of degrees of freedom (DOF) that robots possess are increasing. Thus, they look more anthropomorphic (or zoomorphic) in a realistic way. In addition, the more they become human-like (or animal-like), the more people tend to interact with the robots as they do with humans. The robots need more sophisticated movements close to human or animal like gestures.

In the Personal Robots Group at the MIT Media Lab, Stiehl et al. first proposed a novel robotic Teddy bear platform called, "the Huggable." The Huggable has eight degrees of freedom (DOF), one inertial measurement unit (IMU) sensor, and touch, temperature, and proximity sensors over its full skin. Eight DOFs include two for each arm, three for its neck, and one for ears and eyebrows together.

The Huggable is designed as a robotic companion for use in children's hospitals, elder care facilities, schools, and people's homes with a specific focus on healthcare, education, and family communication. In order to meet its necessities in family communication, education, and health care, it is currently developed to be a semi-autonomous robot avatar that is controlled by an operator who can be any person and may not have a specialty in robotics. Its semi-

autonomous capability helps users think the robot is constantly alive and makes the robot more expressive and responsive to them.

However, controlling a robot is still a challenging task for most people. The Huggable's morphology is too complex to be controlled via a joystick, which is used in many robot platforms that are controlled by operators. For example, it is difficult to control both the Huggable's neck joints and arm joints concurrently. To evoke a simple animation, a user needs to remember different key combinations, the number of which is proportional to the number of different animations. Tele-operation systems other than ones using joysticks are too expensive to be deployable to people's homes and have various limitations as to installation spaces and light conditions.

In this thesis, I developed three different robotic puppetry interfaces with an objective to maximize the robot's communication capacity for its use in family communication, second language learning, and elderly care. We aim to make each interface to be easy to learn and interact with, and also not very expensive so that it can be deployable to people's homes.

First, I explain the software framework of the Huggable. To be semi-autonomous, the Huggable needs several layers of software to process and visualize the sensory information, command behaviors, and control the corresponding motors. We use Microsoft Robotics Developer Studio (MSRS) ("Microsoft Robotics") and the C5M behavior system (B. Blumberg et al., 2002)

as platforms to build our framework. I also explain the common framework that was used throughout this thesis to test the interfaces.

Second, I discuss the design of the web interface. The web interface is a primary interface for the puppeteering in that it displays the field of the robot's vision and the animated 3D robot avatar. In this section, the stale panorama is discussed. It expands the robot's field of vision by taking photos of the robot's surrounding area when it looks around moving its neck. It locates images according to their locations so that the entire view can have a collage of images that spans the viewable area of the robot. An operator can click on the view so that the robot can look to the point directed by him/her. In the aspect of control, it has buttons to evoke sound effects, body gestures, and pointing directions.

Third, I developed two proto-types of the wearable interface. The wearable interface provides an operator sensor bands that he/she can wear on wrists and arms and handheld devices to capture gestures. Two prototypes will include both active and passive sensor technologies to capture human motion data and a software system to both recognize the nonverbal parts of a user's performance and use the captured data to drive the Huggable. The performance evaluation of both interfaces is also discussed.

Fourth, I discuss the design of the sympathetic interface. The sympathetic interface is a physical interface with an appearance that is almost identical to the robot, although its size can be smaller. It contains potentiometers to read joints angles so that when an operator controls the interface bear, the Huggable can

move accordingly. The main idea for the sympathetic interface is for operators to identify (sympathize) the interface with the real robot. An operator can control the interface as if they were to move the robot directly. This came from an idea that people tend to puppeteer a teddy bear by moving its arms while placing it on their lap.

Lastly, the design and evaluation of a human subject study to compare between the web interface and the sympathetic interface will be discussed. The human subject study evaluates two different systems by letting people interact freely with the two interfaces while given few sample dialogues. Subjects also answered questionnaires at the end of the study.

2. Background

Building a robotic puppetry system covers vast research areas such as sociable robotics, tele-existence, artificial reality, embodied conversational agents in immersive environments, human motion capture and analysis, tangible interfaces, and computer animation. Although a robotic puppetry system can comprehensively be included in a realm of tele-existence and tele-operation, it is necessary to study above research areas to bring the level of the system to the state of the art.

2.1. Sociable Robots

According to the survey by Fong & et al. (Fong, Nourbakhsh, & Kerstin Dautenhahn, 2003), past studies on social robots were related to developing insect-like robots, which as a group of anonymous individuals in a society make complex patterns of behaviors and physical structures (Bonabeau, Dorigo, Theraulaz, & NetLibrary, 1999). As researchers become more interested in higher animals like mammals and primates, they became interested in a more individualized society, where each individual matters. Personalized individuals form relationships, networks, and alliances using social interaction and communication skills.

Social robots are defined as embodied agents who can recognize each other, engage in social interactions, possess histories, explicitly communicate with others, and learn from each other in a heterogeneous society of organized humans and robots (K. Dautenhahn & Billard, 1999).

According to Breazeal's taxonomy (Breazeal, 2003), "sociable" robots proactively engage with humans in order to satisfy internal social aims (drives, emotions, etc) and such robots require deep models of social cognition.

The Huggable meets both criteria of defining social (or sociable) robots. However, the puppetry system of the Huggable may look unrelated to its perspective as a sociable robot since it is not considered to be fully autonomous. However, there are many aspects that the puppeteered Huggable can still be seen as a sociable robot.

First, while most of the parts of the robot are controlled via the interface, the other parts still remain autonomous and the social cognition part of the robot will still be maintained throughout the interaction. When puppeteering, its gaze behavior may not be controlled by an interface and driven autonomously. Thus, the Huggable will still be able to run its social cognition functionality while body and facial gestures are being puppeteered via the interface.

Second, the fact that the Huggable is situated in a social interaction does not change and the puppeteering interface is even more strengthening the robot's power to become an actively engaging social companion in the situations mentioned. The puppeteering system enables a human user to effectively

intervene in a social interaction between the autonomous Huggable and a baby (or a child). The Huggable, as an autonomous self, may even learn by itself how to interact with humans in a better way while watching the interaction between a human observer and a human operator. It may also contribute in providing richer expressions to a human than any autonomously driven cases. Thus, the puppetry interface of the Huggable can contribute to many application areas of socially interactive robots such as education, therapy, and entertainment.

2.2. Interfaces for Tele-existence

Tele-existence (Tele-presence) consists of technologies enabling a user to feel his/her presence in a different location from his/her true location. Such technologies include vision, sound, and manipulation. Teleoperation focuses on manipulation that enables a user to control a robot to touch and move objects in remote places. Tele-existence has many application areas such as teleconferencing, operations in hazardous environments, remote surgery, education, entertainment, art, and travel (“Telepresence - Wikipedia, the free encyclopedia”).

Tele-existence and tele-operation systems are already applied to various industrial and military applications.

Among the research utilizing the anthropomorphic robot platforms, tele-operation systems of Robonaut and Geminoid HI-1 can be representative

examples (Goza, Ambrose, Diftler, & Spain, 2004; Sakamoto, Kanda, Ono, Ishiguro, & Hagita, 2007).

NASA's Robonaut used a helmet display, sensors tracking body position and motion by PolhemusTM, and finger sensing gloves like the CybergloveTM by Virtual Technologies, INC (Goza et al., 2004). Positions of sensors and measures of joint angles from the sensors directly mapped into the robots DOFs. To decrease the discrepancy between the data glove and the robot's real hands, a technique relating to inverse kinematics (IK) was adapted. Robonaut's tele-operation system is characterized by its use of commercially available sensor systems and the direct control mechanism between interfaces and Robonaut. However, wearing a helmet and wired devices makes the whole device uncomfortable to be used in people's homes considering the amount of time that a user has to spend on wearing such devices.

Sakamoto's tele-operation systems (Sakamoto et al., 2007) differed from the Robonaut's tele-operation system in that it adapted the semi-autonomous behavior controller to decrease the cognitive load of a human user. The behavior controller consists of a user interface and a state machine. Each state contains a repeatable sequence of actions. Examples of states include Idle, Speaking, Listening, Left-looking, Right-looking, File-playing. File-playing state is used for playing recorded speech files. While in the "Idle" state, the Geminoid can turn his head to left or right, or slightly bend his head. Most of state changes are made manually

through the button-based user interface. Few state changes are made autonomously (e.g. from File-playing to Listening).

In Sakamoto's system, lips were the only part that was synchronized with its human user. He used four infrared reflective markers to capture lip movements via Vicon cameras, and movements were then synchronized to the Geminoid's lip movements (Sakamoto et al., 2007).

The tele-operation systems of AIBO and Quasi are much simpler than the above cases. A user can see a robot's vision through a small video window on a screen and trigger a certain motion by clicking a button among a set of all possible animations.

In a virtual avatar's case, Tartaro's authroable virtual peer (AVP) (Tartaro & Cassell) has a Wizard of Oz (WoZ) user interface to control a virtual avatar (peer). Similar to Quasi, the AVP's WoZ contains different sets of buttons for utterances and body postures for various story sets.

Other puppetry systems will be discussed later in this chapter.

2.3. Human Motion Capture and Analysis

Human motion capture and analysis plays a key role in a robotic puppetry system. Without comprehensive understanding of how a human desires to puppeteer an avatar, designing a state-of-the-art puppetry system is impossible.

Moeslund, in his survey paper in 2000 (Thomas B. Moeslund & Granum, 2001), divided the overall human motion capture process into four sub-processes: initialization, tracking, pose estimation and recognition. In this section, recognition will be mostly discussed. Before 2000, Moeslund categorized research relating to recognition into two sub-categories: static recognition and dynamic recognition. As recognition became one of the main interest areas in human motion capture after 2000, it branched out to a larger number of subfields: action hierarchies, scene interpretation, holistic recognition, recognition based on body parts, action primitives and grammar (T. B. Moeslund, Hilton, & Krÿger, 2006). This taxonomy can be found in his survey in 2006, which can be considered as a sequel to the previous paper.

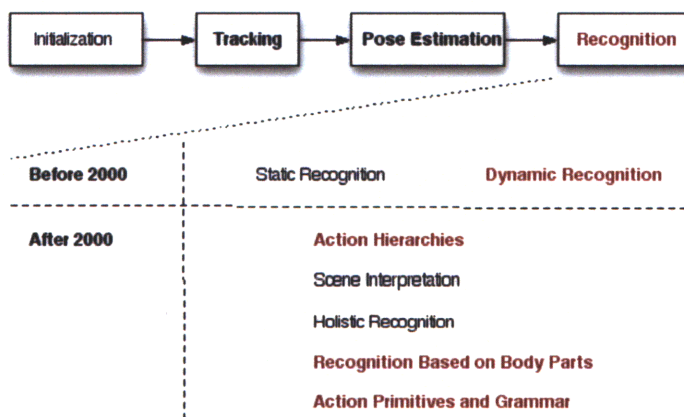


Figure 1. Four Phases of Human Motion Capture and Taxonomy of Recognition (Thomas B. Moeslund & Granum, 2001; T. B. Moeslund et al., 2006).

The interesting thing is that Moeslund analyzed two preliminary works (Wren & Pentland, 1999; Bregler, 1997) on defining and recognizing action hierarchies and its primitives, and predicted that they can be a future direction of research and

it actually became two sub-areas of recognition after 2000 (T. B. Moeslund et al., 2006). Many advances have been made since then. However, defining action primitives and extracting them from a large number of collected action sequences still remain to be explored. While there are already many published papers on preliminary action primitives such as running, jumping, walking, and running, the number of research results are much less in understanding more sophisticated human actions such as understanding the scene context and interaction with other humans (T. B. Moeslund et al., 2006).

Both tracking and pose estimation are also important for controlling an avatar. Without assuring robustness in tracking, accurate pose estimation cannot be guaranteed.

Among various research results on control applications, Moesland stated that Wren's work can be considered as the state of the art (Thomas B. Moeslund & Granum, 2001). Wren's tracking algorithm called pFinder is special in its use of a human model and kinematic constraints to obtain 3D position and orientation estimates of the hands and head and the Kalman filter model to track blobs robustly (Wren, Azarbayejani, Darrell, & Pentland, 1996).

In understanding gesture, researchers have drawn ideas from speech recognition techniques to use Hidden Markov Models (HMMs) which are currently mostly wide spread among researchers in this area. The early use of HMMs can be found in many past works conducted by various researchers

(Bregler, 1997; Schlenszig, Hunter, & Jain, 1994; Starner, 1995; Wren & Pentland, 1999; Yamato, Ohya, & K. Ishii, 1992).

2.4. Embodied Conversational Agents

We can see virtual avatars as semi-autonomous agents, which are designed to communicate with humans and have intelligence to interact like humans by sensing the environment and choosing the right actions to take in an anthropomorphic way. In that sense an autonomous agent, which is anthropomorphic in its appearance and has similar expressive power to the avatars, is almost identical to a robot except in its non-physical existence. These agents are called “embodied conversational agents (ECA)” (Cassell, 2000). Cassell and her colleagues’ work include understanding expressive power of a human and finding ways to implement communicative abilities of humans in an agent. Her system displays an anthropomorphic figure on a screen using three-dimensional computer graphic technology with various multi-modal interfaces.

Her work in embodied conversational agents includes understanding verbal parts of communication, nonverbal parts of communication, and applications which place an agent in a pedagogical situation for children. The most important components of communication, when a human is interacting with a human face-to-face, are spoken language and nonverbal languages. Spoken language can be transmitted via the voice-tunneling feature implemented in the Huggable remote control interfaces. Nonverbal language is more of an issue.

Nonverbal language comprises body and facial gestures. She categorized gestures into three categories: emblems, propositional, and spontaneous gestures. She also subcategorized spontaneous gestures: Iconic, metaphoric, deictic, and beat gestures (Cassell, 2000). This categorization will be helpful in better understanding different types of gestures, which do and do not need to be recognized through an interface.

In developing animated pedagogical agents, it is necessary to put an agent in a specific situation and write a sequence of interactions and agent's according behavior through the interaction. Tartaro's work on children with autism shows a good exemplar of applying a system in a specific environment. It also shows possibilities of the Huggable applying to pedagogical situation with children with autism (Tartaro & Cassell).

2.5. Computer Animation and Computer Puppetry

2.5.1 Computer Puppetry

Computer puppetry brings computer-generated (synthetic) characters to life by translating a human performer's live motion to the movements of the character in real-time (Sturman & Sturman, 1998).

Real-time puppetry systems always include motion-sensing technologies to capture a human performer's motion data. In early 70s, Polhemus first developed electromagnetic sensors to track the body motion (Raab, Blood, Steiner, & Jones,

1979). Other wearable sensing technologies use potentiometers (Anaimac, 1962-1969), exoskeletons (e.g., Gypsy 6 by Metamotion), fiber optics (e.g., ShapeTape by Measurand), a combination of accelerometers, gyroscopes, and magnetometers (e.g., IGS-190 by Metamotion and Moven by MotionSens), and light reflective markers (e.g., Vicon System). Previous works on tracking joint angle using accelerometers, gyros, and magnetometers include Fontaine et. al, Young et. al, and Yeoh et. al's work (Fontaine, David, & Caritu; Yeoh et al., 2008; Young, Ling, & Arvind, 2007). Non-wearable systems mostly use current computer vision technologies to track a user's body parts, retrieve body postures, and recognize body gestures.

One of the early systems of computer puppetry includes the ALIVE project (Maes, Darrell, B. Blumberg, & Pentland, 1997) by Maes & et al. The current behavior system of the Huggable also descended from the C5M architecture (B. Blumberg et al., 2002), one of early examples of synthetic characters by Blumberg & et al. In his project "(void) **", Blumberg used Benbasat's inertial measurement units (IMU) (Benbasat & Paradiso, 2002) to implicitly drive avatars inside the virtual world.

Bobick and colleagues also developed an interactive playroom environment called the Kidsroom (A. F. Bobick et al., 2000). In their environment, they used a technique called temporal templates (Davis & A. Bobick, 1997) to understand overall actions sequentially performed by a human user. Actions understood were used when the cartoon character, "Monster", mimicked the human users' motion.

What makes computer puppetry different from the motion-capture technology used in movie industries is consideration of real-time constraints. Shin et al. (H. J. Shin, Lee, S. Y. Shin, & Gleicher, 2001) provides a novel inverse kinematics solver, which consider tight real-time constraints. His solver considers cases where different sizes and proportions between a human and an avatar matter. Along with the retargeting issues of adapting captured human motions to motions of a new character, noise and glitch issues were considered together with the Kalman filter methods.

2.5.2 Motion Graphs

In the most rudimentary (or current) computer animation platforms, even in some of the most advanced video games, animations played in virtual avatars are played from a beginning to an end. This makes a sudden change in a avatar's action difficult. In a commercially available virtual environment like Second Life, if you evoke a certain animation sequence, the current playing sequence immediately stops and continues to the next sequence. Such a case is improbable in case of robotic platforms since you cannot jump from one position to another.

To solve the problem, researchers in computer graphics have developed the notion called the motion graph. The motion graph is a directed graph where a node represents a frame (a pose in an avatar) in an animation sequence and an edge represents the next frame that the current frame can go to (without any major changes in values of joint angles). The motion graph also connects edges to

similar nodes in different animation sequences. Only the next frame in a given animation can be played after the current sequence, but in case of the motion graph, a state can change from the middle of an animation sequence to the middle of an animation sequence. Many different techniques have been developed to find similar poses from independently made or collected animation sequences (an animation sequence can be made by an animator or collected from a motion-capture system).

Currently, the Huggable has no such mechanisms. Instead, the Huggable's behavior system (C5M) has the pose-graph mechanism (Marc Downie, 2000) that the start frame and the end frame of each animation have independent poses and such a start pose and an end pose are connected to other animations' start and end poses.

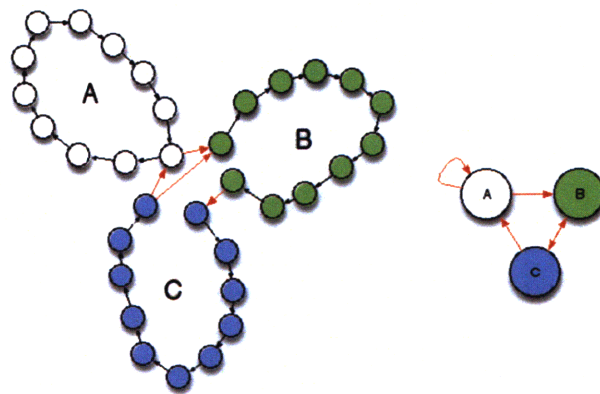


Figure 2. The pose-graph of the C5M's motor system. Each node on the left graph represents frames (poses) in an animation sequence. Different colors represent different animation sequences. Directed edges with the color red refer to the connections between different animation sequences. The diagram on the right shows the abstracted form of the pose-graph on the left. An animation. B can be played after A or C. C can be played only after B. A can be played only after A or C.

2.6. Tangible Interfaces to control virtual and physical avatars

The sympathetic interface that is explained later in this thesis borrows many concepts from tangible interfaces in the past. Unlike the traditional GUI (Graphical User Interface) that uses a mouse and a keyboard, tangible interfaces connect everyday physical objects into digital information (H. Ishii & Ullmer, 1997). Each object or a part of an object may contain digital information. In the case of puppeteering, we can think of an interface that has a physical structure of a robot and function as an input device for control. The physical configuration on each DOF in the interface transforms into a joint position through a sensor. The information extracted becomes digital.

One of the earliest tries of a tangible interface to puppeteer a robot was conducted by filmmakers (Esposito, Paley, & Ong, 1995). To make key frames in the movies, they used an interface called “Monkey” in measuring articulated poses of a virtual human. It looked like an articulated artist’s doll and had a total of 35 sensors in its body and three additional ones attached to the stand. A human user changed the articulation of the interface each time, he/she want to input a key frame to an animation. Then, key frames were interpolated to create a sequence of a full animation. Nonetheless, it was not used as a real-time input device.

The word “sympathetic interface” was first used in Johnson et al.’s work (Michael Patrick Johnson, Andrew Wilson, Bruce Blumberg, Kline, & Aaron Bobick, 1999). They developed a chicken-like plush toy that had various sensors

to detect its behaviors controlled by a human operator. A squeeze sensor for its beak, flex sensors for its wings, and potentiometers for its neck were used to control a semi-autonomous virtual character. It mainly acted like a voodoo doll. Although their interface was semi-autonomous, it was to change the course of action into another. It was not meant to drive the avatar directly such as changing its pointing direction and gaze direction.

Posey and TUI3D also transformed an articulated pose of a structure to a virtual structure and they were used to manipulate virtual characters. Both Posey and TUI3D processed the data in real time (Mazalek & Nitsche, 2007; Weller, Do, & Gross, 2008).

None of these interfaces were used to control a physical robot. The Stan Winston Studio, specialized in producing robots starred in commercial movies, used articulated structures to puppeteer their robots in the movies. The robots were controlled in real time (“Stan Winston Studio”).

3. The Huggable

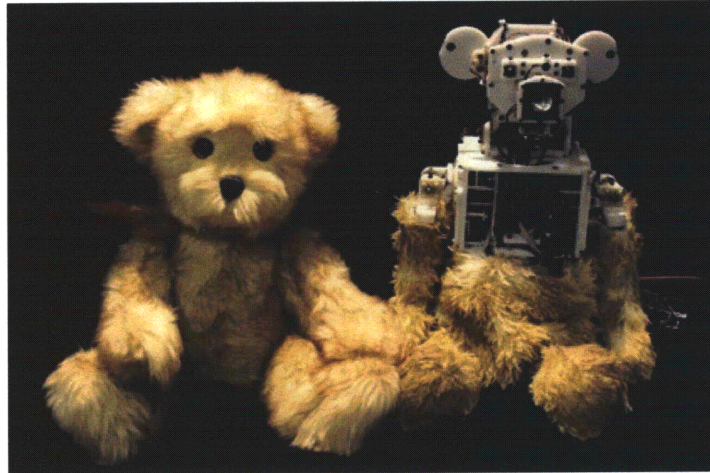


Figure 3. The Current Huggable V3.0 (right, in development) and the Concept Plush Bear (left). The V3.0 Huggable currently does not include the sensitive skin system and soft silicone rubber beneath the fur. When the V3.0 is fully finished, it will look like the plush bear on the left.

In this chapter, we explain the framework of the Huggable that was used as a main platform to test the interfaces throughout this thesis.

The Huggable (Stiehl et al., 2005) is a personal robot that has a teddy bear like appearance. It has capabilities to express itself using motors to manipulate 8 degrees of freedom (DOF) in its arms, neck and ear and a speaker in its mouth. The robot also has various sensing capabilities to recognize different stimuli. It understands people's social touch by using piezoelectric touch sensors and proximity sensors. It picks up its own orientation and movement through an inertia measurement unit (IMU). It can see through the video cameras attached in the eyes and hear from the microphones in its ears.

3.1. Hardware Specification

The Huggable platform has evolved since its debut in 2005. The below list contains all the hardware features that the Huggable currently has in its version 3.0.

- 8 DOFs in the neck (yaw, roll, and pitch), ears (two ears linked together as 1 DOF), and shoulders (rotate and in/out)
- Hybrid belt-gear driven motor system consuming low power to allow longer battery life
- Somatic touch sensors over the full body under its furred skin (this has not been integrated into the V3.0 system in this thesis)
- Color and black/white camera in its eyes
- Microphone array in the head and speaker in the mouth
- Embedded PC with Wireless networking to run the MSRS services to process sensor inputs
- Inertia Measurement Unit (IMU) for body motion recognition
- Passive potentiometers that measure joint angles in hips and ankles

3.2. Autonomous mode VS Semi-autonomous mode

We have built the Huggable to be a robotic companion for people of all ages and assumed it can act on its own. Whenever a person wants to interact with the robot, it is designed to provide appropriate feedback to the person. For example,

children may pat or an elderly person may hug the bear. The Huggable may react to such social touches from a person with its sound effects and body gestures. They can be either a 'giggling' sound with a 'clapping' gesture or playing a 'hugging back' gesture to the elderly person. It may also randomly look around the room from time to time proving that it is awake. When it behaves in such a way, we say the Huggable is being autonomous.

On the other hand, to meet its specific needs in family communication and education, or in a research using the Huggable as a tool, an operator needs to control the robot by him/herself. There can be situations where the robot needs to act based on a sophisticated script that current sensing technology cannot support. Speech recognition has not yet been developed to understand people's every day conversation. Computer vision technology has not been developed to recognize every day objects in people's homes. Moreover, there can be situations where people want to interact through the robot. Children patients may be too scared to talk with their doctors. Children may explain their status of illness more freely to a robot friend whose appearance is affectionate to them. We often see children talking to a doll and tell many stories of theirs. In this case, doctors or nurses may want to interact through the robot.

An operator may not be a specialist in robotics especially when it is used in family communication and education. They can be teachers, parents, or grandparents. They may take full control of the robot, but it also can be a burden for them. As explained earlier, the morphology of the robot is already complex

enough to be controlled by just one person. It can be a cognitively hard task for them to speak through the microphone and control the robot at the same time. For this reason, we build the Huggable to be semi-autonomous. While it is not in control by the operator, it acts on its own as if it is in autonomous mode. It may look around the room or respond to different stimuli. However, when the operator commands gestures or evokes sound effects to the robot, it may stop moving on its own and play ordered behaviors.

3.3. Huggable Semi-autonomous Control System Layout

The Huggable has three software systems. The first system processes and visualizes all the data that come from the robot's sensors and the second system decides and controls behaviors of the robot. The third system is a set of three interfaces on the remote side. They are described in the following chapters.

As seen on Figure 4, the first software system resides in two computer systems: the embedded platform in the Huggable and the local MSRS service server on the Windows XP laptop near the robot. They are connected wirelessly via IEEE 802.11abg. The second system runs on a Mac Os X computer and talks with the MSRS Service Server and the Embedded PC through the Ethernet and wireless connection respectively. To communicate between these systems, we use the Inter Robot Communication Protocol (IRCP) developed by the Personal Robots Group (Hancher, 2003).

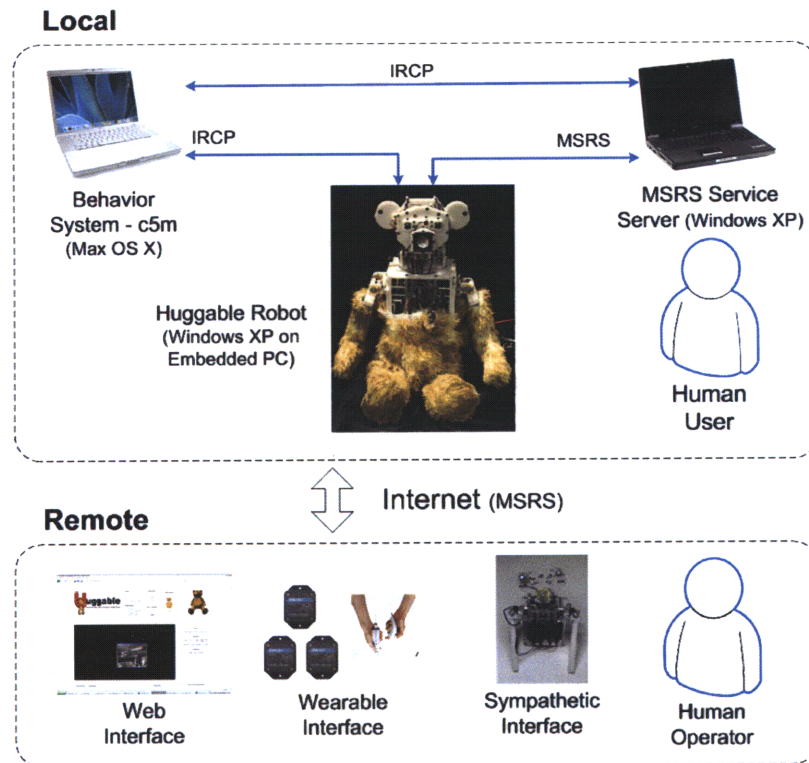


Figure 4. The Huggable’s Semi-Autonomous Control System Layout. On the local side, all the visual, auditory, and somatosensory input from the human user is processed on the embedded PC inside the Huggable and sent to the MSRS service server that visualizes high-level sensor information through the web interface. The C5M system receives inputs from the MSRS server, directs the autonomous behavior of the robot, and sends motor control information to the embedded PC. The human operator on the remote side controls the robot via the remote website. The operator may also wear orientation sensors (3DM—GX1) and hold Wii Remotes to control the Huggable via his or her own gestures or use the sympathetic interface to manipulate the robot’s DOFs.

On the remote side, the operator may use a laptop or desktop. In this thesis we only have developed the system to work on a Windows PC with the Firefox web browser. It talks with the hardware control interfaces via Bluetooth and USB.

3.3. MSRS Services

The first system is to process and visualize the data coming from all the sensor units. We have utilized the Microsoft Robotics' Developer Studio (MSRS) as a main platform to develop necessary software services (Toscano, 2008). MSRS helps to build a cloud of software services to mutually exchange the data. Some services are directly connected via USBs and serial ports to receive data from hardware sensor boards. Among them, certain services have classification algorithms to abstract the data into high-level sensory signals. For example, the PotTemp board sends the passive potentiometer information and the temperature sensor information to the MSRS PotTemp service. It collects the data and transfers to the web publisher and the C5M behavior system to notify about changes in the provided information. Every sensory MSRS service sends information to the web publisher so that when it is viewed inside the web browser. The web publisher is a web server providing the web page we call the web interface. It collects the data from all the sensor services and visualizes the data. It is also connected to the behavior system to retrieve the 3D avatar animation feed.

In Figure 5, its overall structure and the connection between components are depicted and in Table 1, the descriptions of each service are listed.

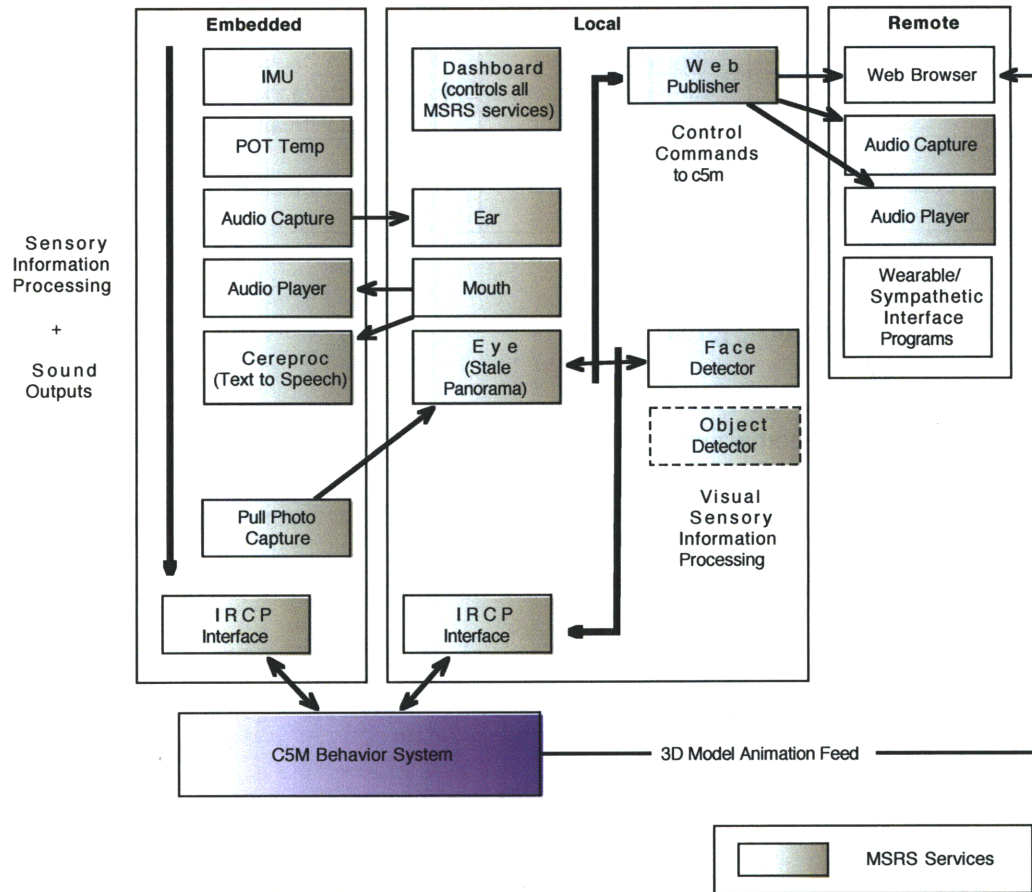


Figure 5. MSRS Service Framework for the Huggable. Sensor services reside on the embedded system inside the physical robot. They, directly connected to the hardware devices, collect and process the data. The local MSRS service server collects the data and process the high level information. Computer vision tasks are processed in the local server so that it will have more computing power than the embedded system that is designed to be efficient in power consumption. The C5M behavior system collects all the data through the IRCP protocol. It also provides the 3D avatar animation feed to the remote web browser.

Table 1. The descriptions of MSRS Services.

Service	Inputs	Outputs	Connected to
PotTemp	Joint angle values from potentiometers in hip joints and ankles.	Joint angles in Radians	C5M (via IRCP Interface) Web publisher
IMU	Raw values from three-axis accelerometer Raw values from tilt sensors	Orientation in three axis, Recognized Abstract Motions	Web publisher C5M (via IRCP Interface)

Pull Photo Capture	Raw video stream	Raw video stream	Eye
Mouth	Raw audio stream	Raw audio stream	Web publisher
Eye	Raw video stream	Stale panorama view	Web publisher
Ear	Raw audio stream	Raw audio stream	Web publisher
Face detector	Raw video stream	Locations of faces	C5M (via IRCP Interface) Eye

3.4. C5M Behavior System

Second, to decide and control the behaviors of the robot, we have used the C5M behavior system (B. Blumberg et al., 2002). It receives sensory information from the MSRS services and processes the information to drive the 3D model of the Huggable. Each DOF in the C5M's 3D model is then transferred to the hardware motor board that drives each motor directly.

The C5M behavior system has mainly built to be a simulation environment for virtual autonomous agents. In its system, it constructs a world in which creatures, autonomous agents in other words, can play. Each agent can have a set of sub systems to process the information that come from outside. An agent also can act to influence the environment and other creatures.

The C5M behavior system has five sub system layers (Gray, 2004).

The sensory system is the gateway for all the inputs that come from outside sources. It processes the incoming data and labels them. It also abstracts, combines, or divides the information to make them easily treatable in the upper layer.

The perception system is inspired by the human perception system. It consists of percepts that are fired when a specific type of input comes in. Such percepts

construct a hierarchical structure like a tree. For example, if the Huggable can sense “tickling”, “patting”, “pinching” from its skin, its parent percept can be “touch from the skin”. Since it is hierarchical, a set of all percepts becomes a tree. There also can be derived percepts. Derived percepts are calculated from the existing percepts. If the Huggable is currently sensing a “human” in its vision and its location is changing, then it can make another percept called “human moving”.

The belief system keeps records of percepts and sometimes combines the percepts together to make an entity. For example, if the Huggable has various percepts relating to a human recognized by the sensor devices, you can collect them together and put them in one belief called “Human A”. This belief will be persistent in its system as long as it exists in the perception system. If it does not appear for a significant amount of time in the perception system, it may be culled so that the belief system does not need to keep records of every percept that existed when the robot was on.

The action system governs the overall behavior of the creature. It has two main components: action groups and action tuples. An action tuple has different modules to calculate a trigger condition for an action, to decide when to stop, and to command the motor system and the sound player to play an animation and a sound effect. An action can be triggered when a certain percept or a group of percepts is fired. An action can be stopped when a certain percept or a group of percepts stopped firing. Action groups contain and manage action tuples and decide which action to active in its action group.

While all the other systems indirectly influence the creature, the motor system directly controls the 3D avatar of the creature and manages playing actual animations for the avatar. Most of the puppeteering system resides inside the motor system.

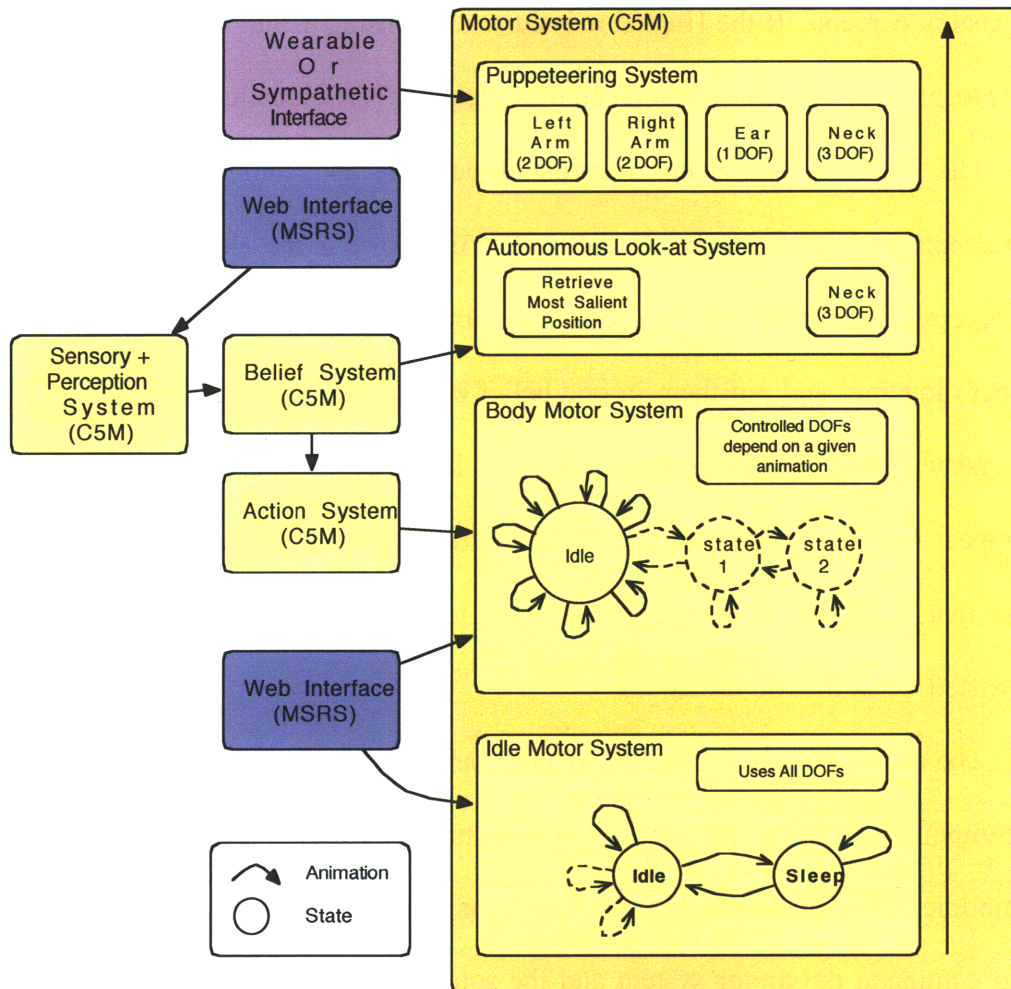


Figure 6. C5M’s Motor System of the Huggable. It has four sub systems: the “idle” motor system, body motor system, autonomous look-at system, and puppeteering system.

As seen on Figure 6, the current motor system of the Huggable has four sub systems. The “idle” motor system takes the lowest priority and plays an “idle”

animation for the robot as a background. Whenever an operator evokes an animation through the web interface, it is directly sent to the body motor system and plays an animation such as “both arms up”. The autonomous look-at system has a higher priority than the body motor system. It decides where to look for the Huggable. When a body animation like “nodding” plays back, the neck of the Huggable first go to a specific position that it wants to look and plays the animation relative to the neck position that it is currently at. The puppeteering system has the highest priority that if an operator wants to control the robot so that it can override all the other systems and give a full control to the operator for a specific part that the operator is moving. Each part is controlled independently so that the other parts can still remain being autonomous and playing back an “idle” animation.

Huggable's Sensory System Layout

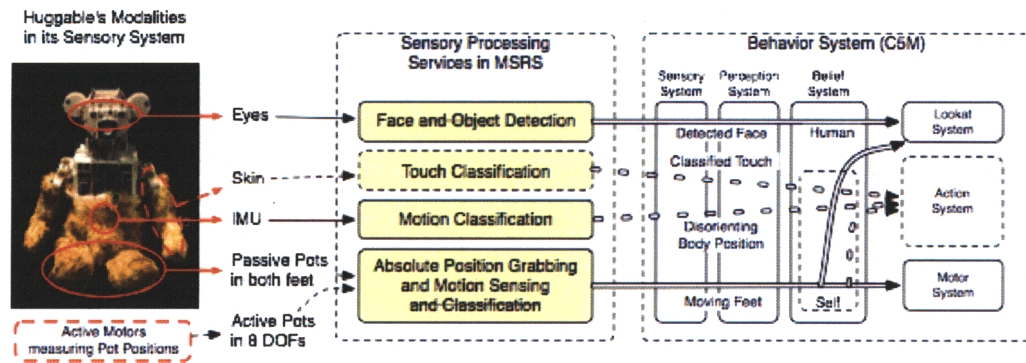


Figure 7. Huggable’s Sensory System Layout. Five sensory inputs are processed through MSRS services. Inputs from passive pots directly go into the behavior system. Changes in DOFs are sent to the motor systems to manipulate the DOFs in the 3D model of the robot. Other inputs are classified in the MSRS services and sent to the behavior system as abstract sensory signals.

Figure 6 shows how each sensory input in the Huggable is processed inside the behavior system. Visionary inputs go to the Look-at system to change the robot's gaze through the belief system and the sensory information that relates to touch and motion go to the action system to change the behavior of the robot.

4. Web Interface



Figure 8. The Web Interface.

The Web Interface (WI) is the primary interface for puppeteering the Huggable. An operator may control the robot through other interfaces, but he/she always need to look into the web interface to watch the robot's current vision and the reflection of the robot (Toscano, 2008). The following sections will cover its main features.

4.1. Situation Awareness

First, it provides situational awareness to the operator. The operator can see what is happening around the robot's space and watch performances of the robot. The Web Interface contains the field of the robot's vision and the real-time

animation video feed showing the robot's 3D model. The operator can see what the robot can see through the view (a black box with a collage of video feed images on the lower left side of the web page) inside the web page and watch the current acts of the robot by looking at its animation reflection (upper right corner on the web page). It is also possible to see passive parts moving by looking at the animation feed. The data from the passive parts (its feet and hip ankles) are collected from the PotTemp board (potentiometer and temperature sensing board) and sent to the C5M system to change the 3D model. With these views, the operator can be aware of the current situation in which the robot is at a remote place.

Stale Panorama

The stale panorama is a collage of images that were taken by one of the robot's eye camera while a robot is moving its neck looking into its surrounding area. The only moving part in the view is the current field of view. It will be inside a yellow box located based on the neck's true location.

The operator can control the robot's gaze by clicking on the view. The blue box tells the place that the operator wants to go to. When he/she clicks on the view, the yellow box will slowly follow to the blue box.

Few seconds after the robot finishes directing its gaze to the position that an operator clicked, it goes back to its random look-at mode and start to look around the room. This is another feature in the semi-autonomous puppeteering mode.

3D Avatar Visualizer

The visualizer inside the web page will show to the operator the current postural state of the robot. It will be a 3D virtual model of the robot that resembles the real robot and the DOFs will be identical to the physical one. It does not only show the active parts moving, but it also shows the passive parts that can be manipulated by the user on the remote side. Potentiometers in passive joints will measure the joint angles and send the information to the behavior system to visualize into the virtual avatar.

4.2. Firing Control Commands and Audio Interface

The web interface provides the ability to control the robot to the operator by using buttons on the web page. Sound effects, basic body gestures, pointing gestures, and emotion gestures can be evoked inside the web site.

Table 2 contains all the sound effect and animation that the current robot can play.

Table 2. The Huggable's Sound effects and Animation

Sound Effects	Head Gestures	Pointing Gestures (Left Arm, Right Arm)	Body Gestures	Emotional Gestures
Hello	Nod	Front	Raise Arms	Happy
Bye-bye	Shake	Middle	Greet	Sad
Giggle	Flick Ears	Side	Goodbye	Confused

Growl	Side to Side		Pick Me Up	
Snore				
Curious				

For sound effects, we use a human voice imitating a bear. For each sound effect, we have five to seven different samples. Whenever one sound effect is evoked, the program randomly chose which sample to play.

For pointing, we have three different animations. The “front” animation means that the robot is pointing straight to the front. The “middle” animation points to the direction which is 45 degrees left or right from the front. The “side” animation points all-the-way to the left and right.

For body gestures, we have arm waiving gestures that the robot waives either its left (“greet”) or right (“bye-bye”) arm for few seconds. There is no reason that we named the gestures “greet” and “bye-bye”.

Emotional gestures include combinations of gestures. For example, the “sad” gesture will slightly drop its chin down and rock its head to left and right while both arms are waving in front of its eyes.

To be able to provide the direct audio transfer service, the web interface has the MSRS audio capture and player services to support this additional feature. The operator can wear a headset and talk through the microphone. It will be directly transferred to the Huggable’ mouth speaker via the MSRS services.

On the other hand, it also provides text-to-speech interface using the CereProc SDK (“CereProc Text to Speech - Homepage”). The operator can type into the text box to make a speech. They can hear back the sound via the headset.

5. Wearable Interface

5.1. Proposed Approach

In the first two proto-types, I developed robotic puppetry systems which directly capture human motion through sensing technologies, process through gesture recognition techniques, and drive a robotic avatar in intuitive and efficient ways. The system consists of hardware sensing devices and the puppetry software framework.

The core goal for building a customized hardware device is to make it as inexpensive as possible to be deployable to general people's homes. One of the Huggable's main design goals is to provide a different and exciting form of family communication. However, if such devices are too expensive like any motion sensing technologies out for commercial use, it will not be possible to deploy them to more than 30 homes for a research purpose.

Hardware includes both active sensing technology and passive sensing technology. Active sensing refers to technologies that obtain motion data both from wearable sensors and sensors around surroundings. Passive sensing refers to technologies that obtain motion data from natural resources such as visible light or electromagnetic wavelength. The drawback of active sensing is that devices are sometimes intrusive to human users. The drawback of passive sensing is that technology depends too much on environmental factors such as light conditions and etc. The motion-captures through reflective markers and high-hue color

clothes are still wearable, but considered as passive sensing technologies since they are less intrusive.

The software system consists of two main sub-systems: the motion-capture system and the motor drive system. The motion-capture system will aim to make use of inputs from both active sensors and passive sensors and receive data from different body parts such as a face, arms, and hands. The motion-capture system does not only estimate the body pose of a user but also aims to recognize the intention of a user's action to efficiently drive the Huggable and lessen the cognitive load of a user. The motor drive system will aim to robustly control the actual physical Huggable. Glitches or noise in the data may be fatal in controlling a delicate robotic system and a safety mechanism should be involved.

In the following two proto-types we tested two sensing technologies (active vs. passive) and two software schemes (gesture recognition based control vs. direct control).

5.2. Recognition using active sensing technology

The first task was to implement a gesture recognition system by adapting HMMs. For this proto-type, one Wii Remote and one Nunchuk were used; they were connected through a wire and used only one Bluetooth connection to a computer. The device sent a total six of acceleration values in three different orthogonal directions for two controllers. The data stream was recorded and used to train six different continuous HMMs for each different gesture that we aimed to recognize. The six gestures were “arms up, arms forward, bye-bye (one arm up

and waiving), crying (wiggling both hands in front of eyes), ear flickering (wiggling both hands above the head), and clapping (with both arms forward repeating moving both hands left to right in an opposite direction)”.

Data was collected from only one person for the first day session. For the next day, data was gathered by eight different undergraduate and graduate students who worked in the Personal Robots Group at the MIT Media Lab. They each repeated one session of six gestures seven times. They were given a page of instruction with figures showing the sequence of each gesture and also watched a demonstrator who performed the respective gestures. Even though they were told to follow gestures that were taught by both instruction and demonstration, real participants did not follow exactly what a demonstrator tried to gesture. Each gesture was tagged by the kind of gestures and the number of repetitions when it was recorded. The human participant used the button on the bottom of the Wii Controllers to let the recording program know when to start and stop recording for one performance of each gesture. Eight different sets of data with seven different sessions of six gestures were used to train six different continuous HMMs separately for each gesture.

Baum-Welch’s estimation maximization (EM) algorithm was used to learn parameters of HMMs for each set of the given data (Rabiner, 1989). HMMs were trained using Kevin Murphy’s hidden Markov model (HMM) Toolbox for the MATLAB (“Hidden Markov Model Toolbox for Matlab,” 2005). To utilize learned parameters for each different HMM, the forward algorithm was re-

implemented to the C language and used for real-time classification. The forward algorithm implementation was based on Kevin Murphy's Toolbox ("Hidden Markov Model Toolbox for Matlab," 2005). The forward algorithm calculates likelihood probabilities of all six HMMs for the given data and chooses the most plausible hypothesis (gesture). Classified data is sent to the behavior control system (C5M) of the Huggable to trigger the respective animations. However most of the gestures were classified correctly on varying human users, "arms forward" and "arms up" were sometimes misclassified, and there were too many false positives for "ear flickering".

Table 3. The algorithm for choosing the most likely action sequence.

<ol style="list-style-type: none">1. If the button is newly pressed,<ol style="list-style-type: none">a. Initialize HMMs for each different gesture (total six gestures)2. If the button is being pressed,<ol style="list-style-type: none">a. Iterate HMMs by feeding accelerometer values into each HMM Likelihood values for different HMMs are calculated.3. If the button is released,<ol style="list-style-type: none">a. Compare between likelihoods of all HMMs.b. Choose the most likely motion sequence.

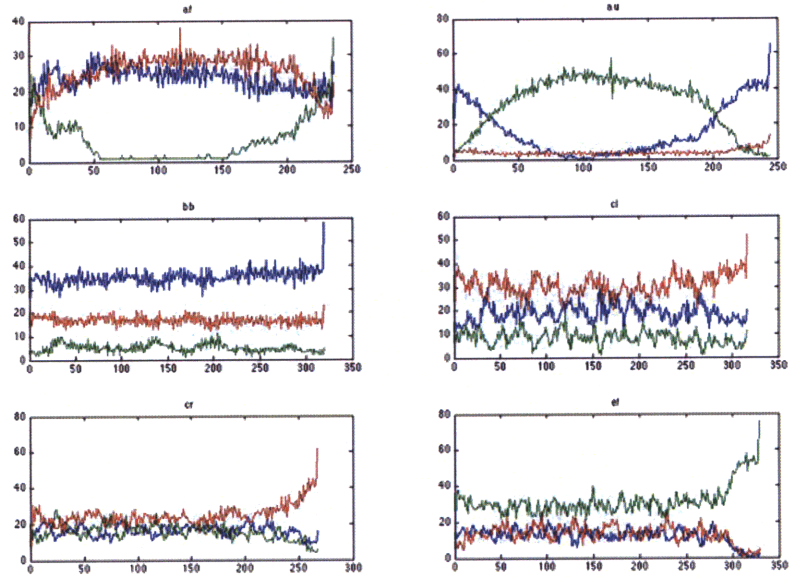


Figure 9. State change histograms for six different gestures.

5.3. Tracking and Pose Estimation

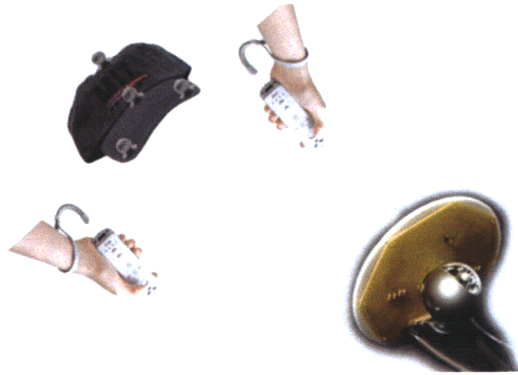


Figure 10. The device setup for direct puppetry system proto-type.

The second proto-type was to test passive sensing technologies and direct manipulation. Moreover, in this second proto-type, hardware of the Huggable's arms were also tested.

For the direct manipulation, it was decided to control both shoulders and the head. To track the head's movements, a baseball cap with four reflective spherical markers in a shape of tetrahedron was used to extract three joint angles, yaw, roll, and pitch, for the neck.



Figure 11. A human user is operating the Huggable through the passive sensing devices.

Most current marker-free body detection systems are not robust enough to the noise and the changes of light conditions. To exclude such problem, the infrared camera and the infrared light reflective markers were chosen instead. For the infrared camera, the Logitech Quickcam Pro 5000 was used; the lens with infrared filter was replaced with a 2mm wide-angle lens, and visible light blocking filter made from exposed color film was attached in front of the lens. Instead of high-resolution stereo cameras, a regular web-cam with the 640x480 resolution was selected. The real resolution used in the software to minimize the latency of

processing was 320x240. The web-cam is modified only to accept infrared lights. Most marker-free detection algorithms that can track hands and head postures without any markers are not robust to changes in light conditions. People's homes are more vulnerable to such changes than environments of research labs. Images of infrared light reflective markers are noiseless and stable.

The image captured from the customized web-cam contained six blobs of a different size and location. To track blobs, the openCV library by Intel was used. After the locations of different blobs were acquired, the algorithm described in Table 3 was used for indexing unordered blobs into six different categories: head top, head middle, head right, head left, hand right, hand left.

The POSIT (Pose from Orthographic and Scaling with Iterations) algorithm was used to calculate the data (Dementhon & Davis, 1995). The POSIT algorithm is modified to adapt the perspective projection assumption from the version of a POS algorithm (Ullman & Basri, 1991; Tomasi & Kanade, 1992). The POS algorithm assumed orthographic projection and provides a poor result when the object is near from the camera which is the case for this puppeteering system. The description for the algorithm is in Table 4.

To track hand movements, for the second prototype, two Wii Remote controllers with reflective tapes on its front were used. However, for this direct puppeteering interface, accelerometers were not used.

Table 4. Blob Tracking Algorithm

<ol style="list-style-type: none">1. Find blobs using the openCV Library2. Initializing the blob order according to the sequence below.<ol style="list-style-type: none">a. Find two hands on the bottomb. Find top two head blobs on the topc. Remaining two blobs become head right and head left blobs.3. Traverse through all possible permutations of blob orders<ol style="list-style-type: none">a. Calculate the sum of square root errors between the current blobs and the last present blobs. Consider cases for<ol style="list-style-type: none">i. When the blob is just disappeared.ii. When the blob remains off-screen.iii. When the blob reappeared.iv. When the blob constantly remains on the screen.b. Exclude Impossible Cases (left and right hand switched, out of bound on calculated).4. Select the most probable candidate (the one with the least squared error) as the next blob sequence5. Calculate joint angles from the selected candidate using the POSIT algorithm

Table 5. POSIT Algorithm (Dementhon & Davis, 1995)

<ol style="list-style-type: none">1. The distribution of the feature points on the object is known and the images of these points by perspective projection are given2. Build scaled orthographic projection (SOP) images of the object feature points from a perspective image. Apply the POS algorithm to these SOP images and obtain an exact object pose (translation + rotation).3. After applying POS to the actual image points, obtain an approximate depth for each feature point and position the features points on the lines of sight at these depths. Then compute a SOP image.4. At the next step apply POS to the SOP image calculated in the above process. Iterate this process until there are no changes. <p>* POS algorithm is used when the orthographic projection is assumed. In the above case, the algorithm has been modified to meet perspective projection assumption.</p>

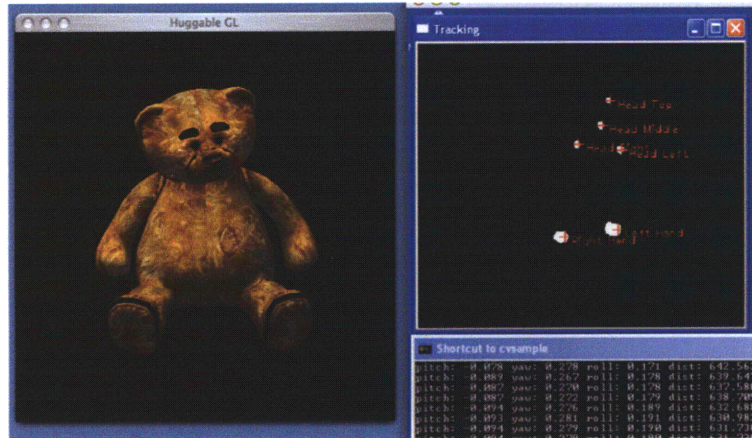


Figure 12. Blob Tracking Software.

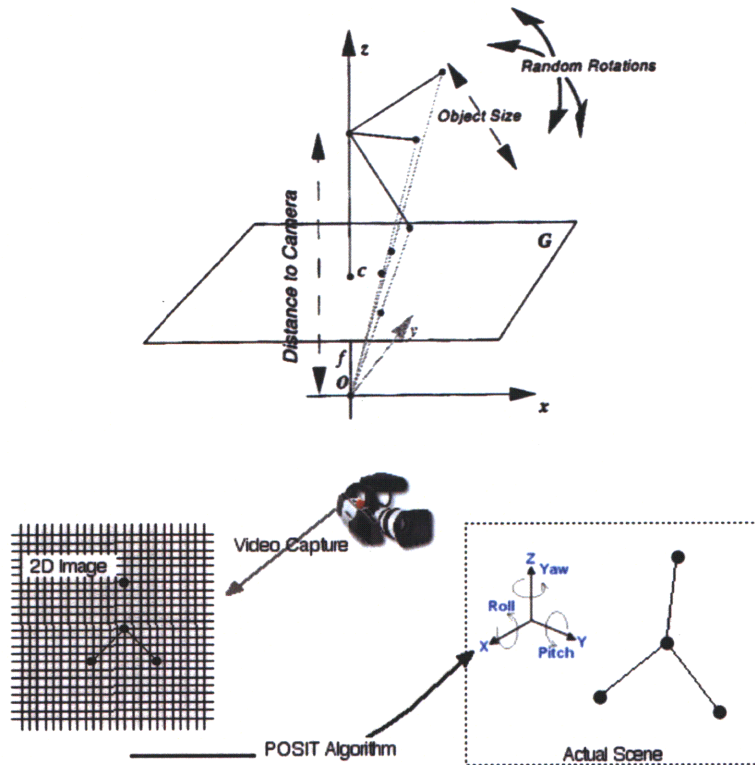


Figure 13. POSIT Algorithm (Dementhon & Davis, 1995).

Joint angles were sent to the behavior control system of the Huggable. The three-dimensional avatar of the robot moved accordingly and motor potentiometer position were sent to motor control software in a series. Motor controller software controlled hardware arms of the Huggable v3.0: the left and right shoulders with 2 DOFs, and the neck with 3 DOFs were built as the Huggable v3.0.

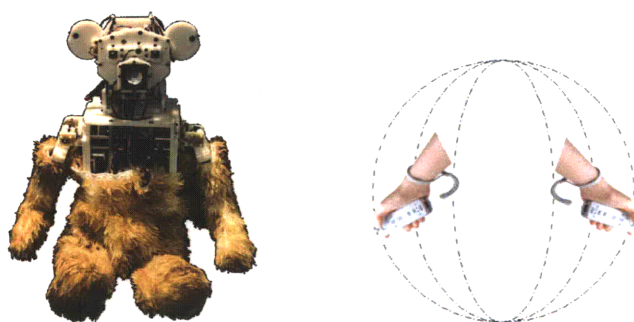


Figure 14. The hardware proto-type for the arms of the Huggable 3.0 (left) and the working range for both arms (right).

5.4. Evaluation of Early Prototypes

To evaluate the efficacy of the system, I measured the latency in two different systems: the gesture recognition based puppeteering system and the direct manipulation system. The latency is the difference in time (seconds) between the user's input and the robot's motion. It was measured by time-stamping the recorded video clips that were captured in a setting that is described in Figure 15 and Figure 16 describes how latency is measured in two different cases showing graphs.

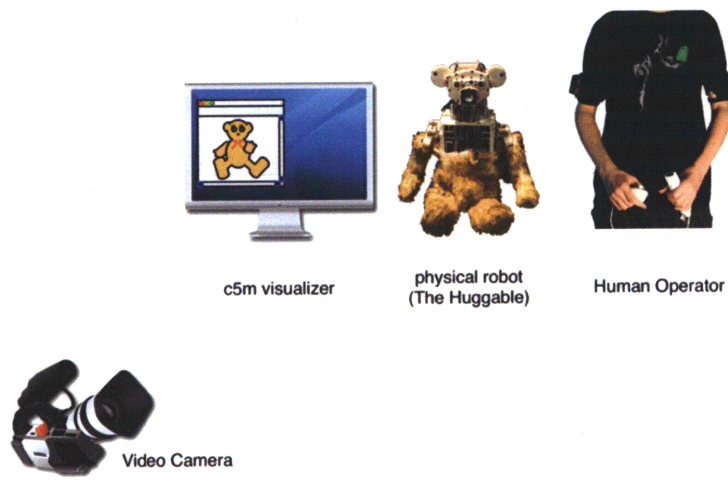


Figure 15. Evaluation Setting. A video camera captures the monitor (showing a virtual model of the Huggable), the Huggable (in its physical form), and the operator. The latencies were measured by time-stamping the recorded video clips. The measured latency is the time between the onset of a motion by the operator and the onset of the same motion on the physical robot’s side.

In the evaluation session for direct manipulation, the operator was told to move his/her arms all the way down to all the way in the middle and all the way to the top. When he/she started his/her action, it was time-stamped in the video clip and the exact time when the robot starts his action was also time-stamped. The difference in time was measured in number of frames (the video clips were recorded in 30 frames per second) and was converted into the scale of seconds. The measured latencies are covered in Figure 17.

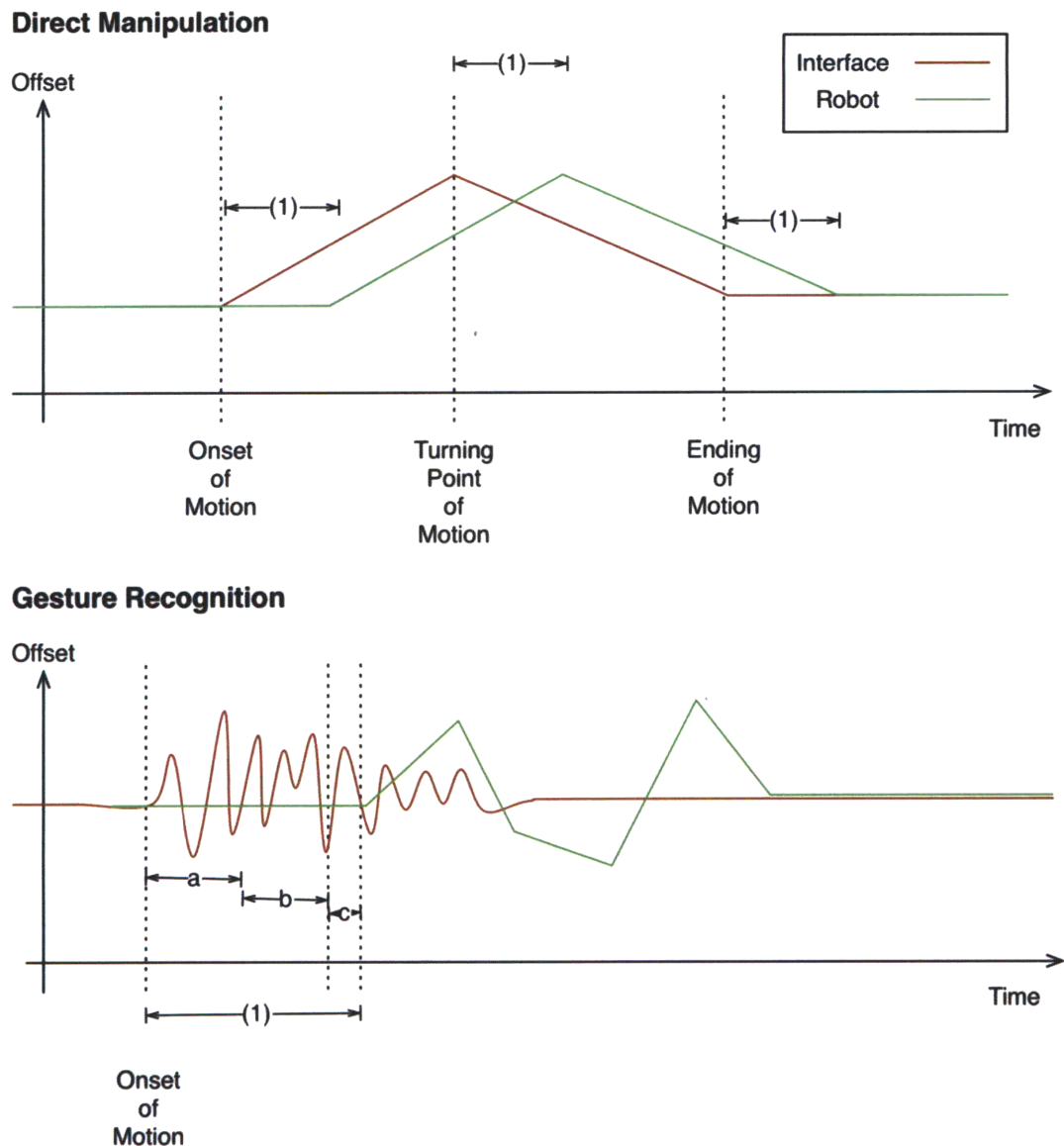


Figure 16. Measure of Latency. The overall latency (1) is measured in two different setups: direct manipulation and gesture recognition. In direct manipulation, the whole action was divided into three steps: onset (when operator starts to move the arm from down to the middle), turning point (when the operator starts to move the arm from the middle to all-the-way up). In gesture recognition, the latency was divided into three different paces: the recognition latency (a), the animation latency (b), and the network latency (c). The recognition latency refers to a delay in recognizing the gesture performed by the operator and the animation latency refers to a delay in evoking a certain animation that is mapped to an operator's specific motion.

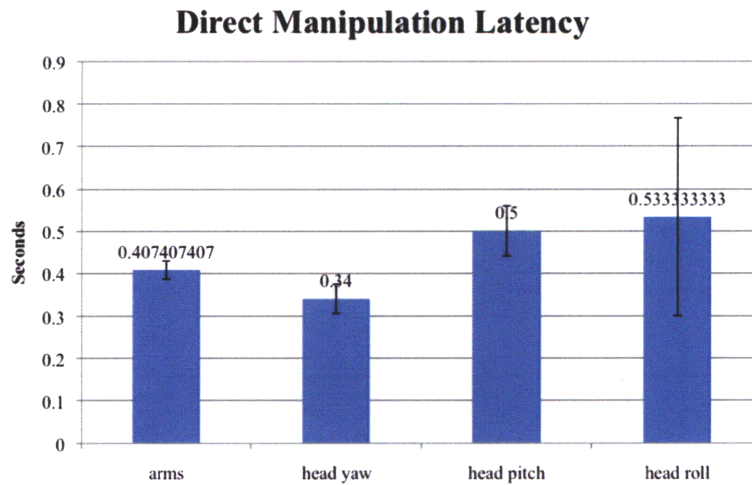
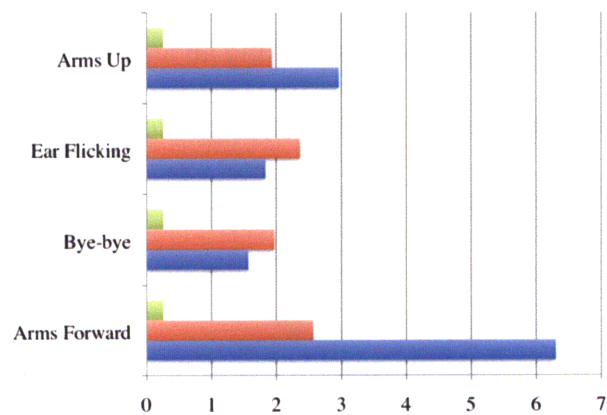
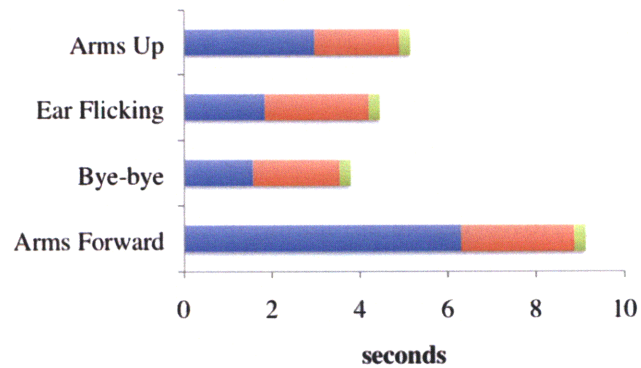


Figure 17. Latencies in direct manipulation. The latency of each joint was measured separately except the arms joints. The latency depends on the adjusted motor speed of each joint and other factors such as network speed and system processing time.

In case of gesture recognition, the latency were divided into three categories: network, animation, and recognition. The recognition latency is a delay between the onset of a motion on the user’s side and the time it was being recognized by the system. The animation latency refers to a delay between the time when the gesture was recognized by the system and the time when the robots initiated its motion. Network latency refers to a delay in traffering data through the network (in this measurement it was set to 250ms by default). This was due to the fact that it cannot be tested correctly unless they are physically located in different places which need to be far enough to measure the latency.

The detailed results are represented in three graphs in Figure 13. The overall average latency in gesture recognition was 5.6292 seconds. Although 5 seconds will not be critical in initiating each different gesture seperatly, it may be crucial

when an operator attempts to tell a story through the interface as it will involve consecutive gestures which need to be recognized continuously by the system. A new algorithm to detect a gesture in its early phase needs to be developed to improve the system. However, the animation latency is also a critical factor in the overall latency. Currently, it takes about 2.2 seconds to evoke an animation. In most cases, it is almost as long as the recognition latency. It is mostly because the system set up more than two phases in evoking an animation. By default, the robot plays an “idle” animation and whenever it needs to evoke a new animation, it waits until the default animation finishes. This process increase the delay and the preparation phase in the animation itself also increase the delay as well.



	Arms Forward	Bye-bye	Ear Flicking	Arms Up
network	0.254	0.254	0.254	0.254
animation	2.567	1.967	2.367	1.933
recogniton	6.3	1.567	1.833	2.967

seconds

■ network ■ animation ■ recogniton

Latency in Gesture Recognition

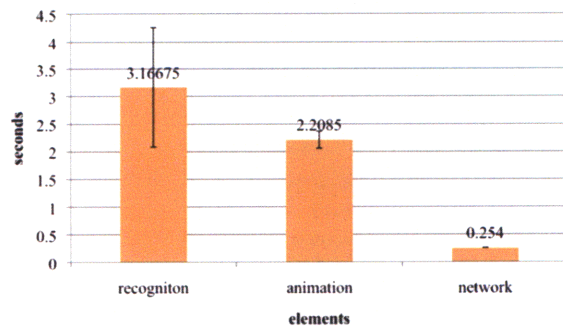


Figure 18. Measure of latency in gesture recognition. Four different gestures were measured and represented in three different graphs. The first graph compares the overall latency in each gesture. The second graph compares the latency in each gesture in three different categories. The last graph compares the latencies in three different categories.

Comparison between two separate modes

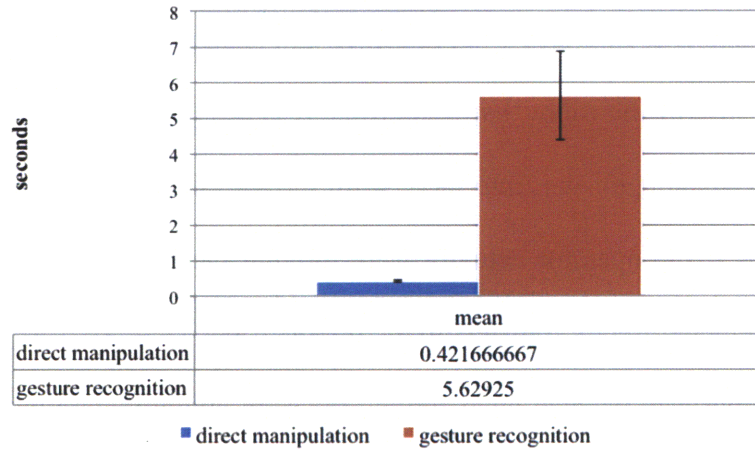


Figure 19. Comparison between two separate cases: direct manipulation and gesture recognition.

The comparison between the direct manipulation and the gesture recognition is represented in Figure 19. The advantage of direct manipulation system is obvious. However, in the direct manipulation mode, the operator needs to move his/her own body parts constantly and it can be sometimes burdensome to the operator.

6. Sympathetic Interface

The sympathetic interface is an interface that its appearance and inner structure resemble those of the robot that it controls. It is designed to assist an operator to identify (sympathize) the interface with the physical robot and control the interface as if the operator actually were moving the robot itself. Therefore, the doll's movement directly transfers to the remote robot.

In this interface, we also have applied the semi-autonomous functionality. The operator can change the gaze direction of the robot by turning the interface's neck and the pointing direction by moving the interface's arm. However, he/she may not need to hold the robot all the time to put the robot in a desired posture. When the operator does not move the part such as arms, neck, and ears, it will go back to its original position and play the idle animation. It will happen independently on each part. Therefore, when the neck is in control and the arms are not, the robot's real neck will be only the puppeteered part. This will lessen the cognitive load of an operator when he/she need to look through the screen and talk concurrently.

The sympathetic interface will be situated with the web interface and the operator will be able to see the robot's view through the web interface and check their control via the 3D avatar animation feed of the robot. While he/she will focus on the web interface, the robot may remain in front of the screen or on the lap of the operator.

6.1. Hardware

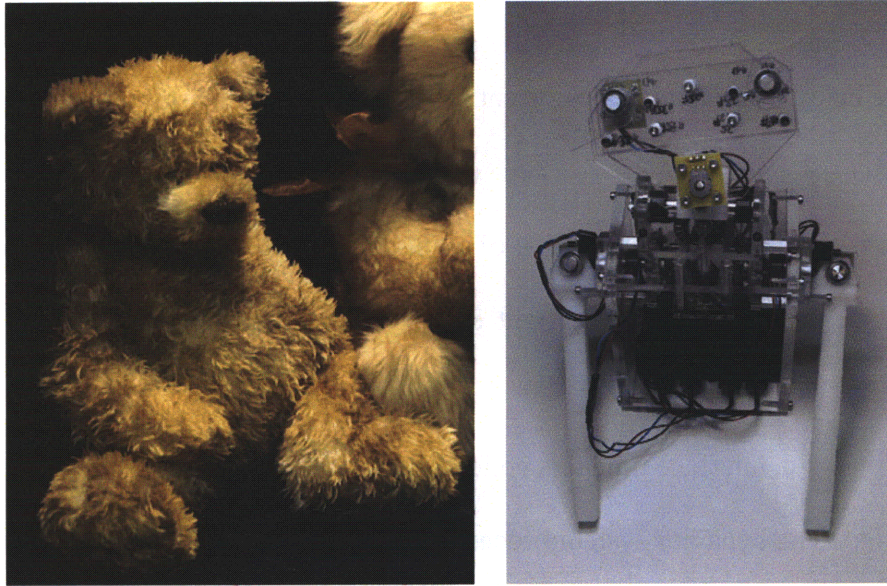


Figure 20. The plush bear (left) and the inner structure of the sympathetic interface (right)

As seen on Figure 20, it mirrors a body of a Teddy bear (left), but has an inner structure (right) that has joints in the neck, arms, and ears. It has 3 DOFs (yaw, roll, and pitch) in the neck, 2 DOFs (up and down, rotate) in each arm, and 1 DOF in the right ear. In each joint, it has a potentiometer to measure joint angles in 360 degrees. It is connected to the hardware interface board that streams position information to the C5M behavior system. In this thesis, we used the inner structure only to test the interface. When it is fully finished it will be stuffed inside the Teddy bear doll and may also be much smaller. Currently, its height is

9 inches and its width is 6 inches. It is small enough for an operator to place on his/her lap.

6.2. Features

The below list explains the current capabilities of the interface.

- The interface has a total of eight potentiometers to measure joint angles in the neck (three), right ear (one side only), and both arms (two for each)
- The interface transfers collected data to the behavior system (C5M)
- The behavior system processes data in such a way that movement of the actual robot resembles that of the interface.
- The behavior system turns on and off the puppeteering feature for each body unit. The body units are divided into the neck, the left and right arm, and the ears. For example, if an operator only moves the yaw DOF of the neck, she/he will take the entire control of the neck.

In the future, the sympathetic interface will not be passive all the time. Although it does not yet contain any actuators, it may later provide an operator feedback when a person next to the robot blocks one arm and it cannot move to the direction ordered by the operator. The feedback mechanism can be either a vibrator or another form.

6.3. Software Specification

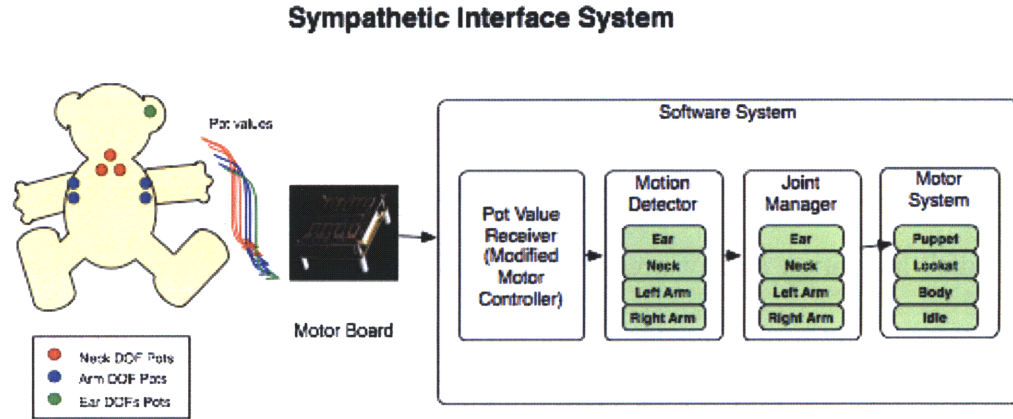


Figure 21. Sympathetic Interface Information Flow Diagram. Position information from the potentiometers are sent to the software system via the hardware motor board. It has four layers to process the information: the information receiver, motion detector, joint manager, and motor system.

To support the semi-autonomous mode in puppeteering, the sympathetic interface has four layers of software components that process the position information from the sensors in the interface. It is currently all embedded in the C5M behavior system.

The main feature that the sympathetic interface has is that its parts are independently controlled. As explained earlier, the interface divides the controlled parts into four: the ear, neck, left arm, and right arm.

After the behavior system receives the position data from the motor board, it calculates differences in values and detects whether the operator is actually moving a specific part or not. Motions in those parts are detected independently.

Second, the joint manager gradually turns on and off each part whenever there is a movement in those parts. Therefore, if the operator starts to raise and waive the left arm of the robot, the left arm of the physical robot will slowly follow the current left arm position of the interface and also follow its movement as well. However, while this is happening, all the other parts will remain playing the “idle” animation or any according animations that the behavior system directs. This enables an operator to control the web interface to change the gaze of the robot while he/she holds the left arm of the sympathetic interface to make the robot point or gesture.

As described earlier in Chapter 3, inside the motor system of the C5M, puppeteering takes the highest priority in governing the motor controls. Whenever there is a movement in the interface, puppeteering will take over the control of the robot for each part.

7. User Study

7.1. Study Design

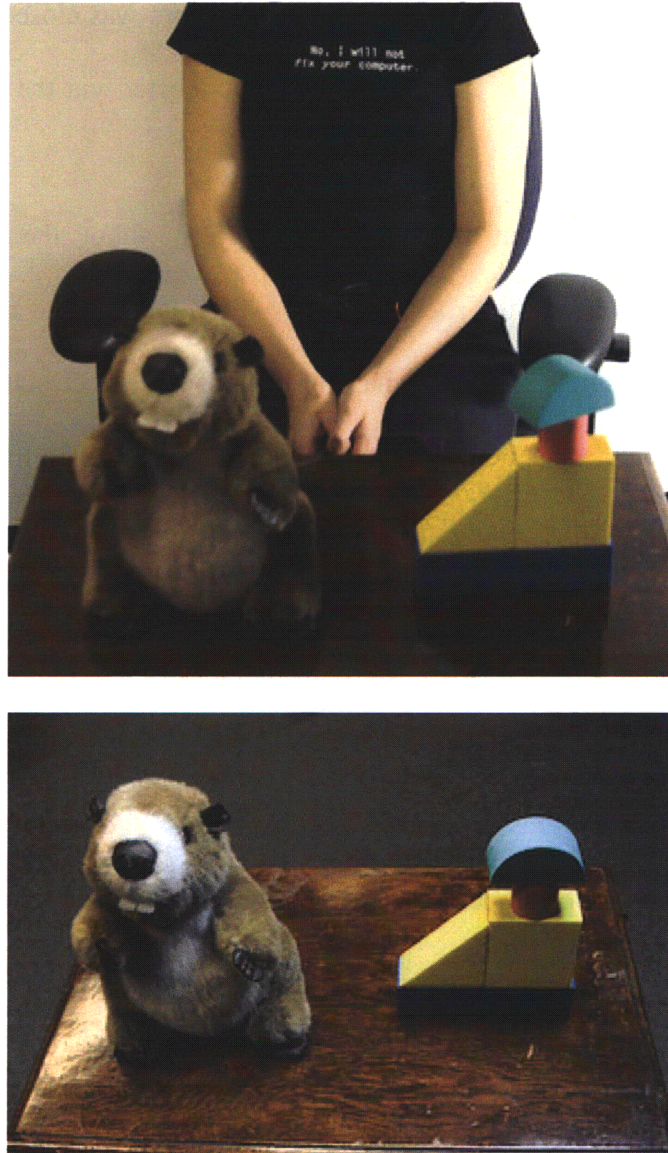


Figure 22. Setting for the study. A beaver doll and blocks were set up on a table in front of the physical robot and an experimenter sat behind the table.

The first study tested the web interface. In this study, an experimenter was on the robot's side to give feedback to a human operator and the subjects' role was to

control the robot as the operator. The robot and the local server were located in the lab's common area and the remote computer was set up in an office room that was 30 feet apart from the robot. The door of the room was closed so that he/she cannot hear outside. All the components were connected via the wired Ethernet for robustness.

Each subject was given five minutes to learn about the interface. The experimenter came inside the room together with a subject and explained about the web interface. The stale panorama interface, sound effect buttons, animation buttons were explained sequentially. After this process, the experimenter stayed in the room to provide suggestions to subjects for interactions.

On the robot's side, there was another experimenter who reacted to interactions with the Huggable and a table was set with two different types of toys: a beaver doll and blocks (see Figure 22).

Thirteen subjects participated for this study and their age range was between 18 and 65. Among thirteen subjects, seven were male and six were female.

7.2. Dialogue

Subjects were given suggestions on a number of things that they can play with the robot. The suggestions were given in the following order.

1. Greetings

Suggestion: Hi, my name is Huggable. How are you feeling today?

Experimenter's answer: I'm fine. How are you?

2. Scratching the foot

Suggestion: My foot is little itchy today. Could you scratch it for me?

Experimenter: (H/she scratched the foot and the 3D avatar's feet was moving in the web interface)

3. Hugging

Suggestion: Can you hug me?

Experimenter: (He/she hugged the robot, the human operator was able to see through the stale panorama view)

4. Putting a hat on the robot

Suggestion: Can you put a hat on me?

Experimenter: (He/she found a hat on the table and put it on top of the robot's head.)

5. Pointing

Experimenter: Would like to point to the toy that you want to play with?

(The subject were given two different types of toys)

Subject: They used buttons to point to the toy they want to play with.

6. Playing with toys (after a subject choose what he/she wants to play)

Suggestion: Can you build a dam with those blocks?

I would like to hug the beaver. Can you bring it to me?

7. Playing different animations and expressing their emotions through emotion animations

Suggestion: I'm very excited. (pressing a "happy" animation button)

8. Good-bye

Suggestion: (Say good bye to the robot) Good bye!

Some subjects followed the experimenter's suggestions, but some made the comments on their own and tried different things.

7.3. Questionnaire

The questionnaires were given at the end of each study. Subjects did not answer the question 3,4, and 13.

1. I feel comfortable controlling the robot
2. I feel confident that I will be able to control the robot in a way that I want.
3. (For the wearable device user only) I feel comfortable wearing the devices.
4. (For the bear doll interface user only) I feel comfortable touching the bear doll.
5. (For the website interface user only) I feel comfortable finding right buttons to evoke different animations.
6. I find controlling the robot's gaze direction (head orientation) most difficult.
7. I find controlling the robot's pointing direction (arm orientation) most difficult.
8. I find controlling the robot's ears most difficult.
9. The voice was delivered clearly to the person who interacted with the real robot.
10. The voice was delivered without a significant delay.
11. I did not feel a significant delay in interaction or response from other person.
12. (For the website interface user only) I did not feel a significant delay in evoking different types of animations for the robot
13. (For the wearable and bear doll interface user only) I did not feel a significant delay in moving the robot's parts (arms and the head).
14. I did not have a hard time finishing the given dialogue (or a given story) in the given time.
15. I feel confident that I delivered what I intended to the person who interacted with the robot which you controlled.
16. I feel confident that the person understood where I was pointing.

9	4.769230769	2.33581762
10	5.307692308	1.86494557
11	5.307692308	2.091321681
12	4.846153846	2.437121343
13		
14	6	2.631174058
15	5.769230769	2.979785375
16	5.692307692	3.251373336
17	5.307692308	3.820131751
18	5.307692308	3.641186861

Most subjects answered the questionnaire positively.

In terms of technical evaluations, there were some problems with the voice transmission and subjects felt less positively about the quality of voice interaction. Some subjects tried to use the text-to-speech interface because the transmitted voice was unclear; they could not hear back their voice well. Some were shy to use their own voice, too. But, most people were not too bothered by the delay in voice transmission. However, they were concerned about the animation delays as mentioned earlier in the wearable case, too.

A father who came with a daughter was very excited to play with the robot. He came up with many stories of his own and wanted to continue playing with the robot much longer. He also used objects near the robot and used them as toys to play with the robot. Among thirteen subjects, two or three people were in thirties and forties who have children. They were more excited and willing to play with the robot than other subjects.

Most people became more interested when they could actually see the robot moving from its eye camera. The 3D visualizer did not help much in providing

the awareness of what current robot is doing. They did not believe the 3D animation video feed more than the video feed from the camera. When they actually saw the arms moving in the video feed, they finally believed that they were actually controlling a robot.

In overall, most people felt it was not difficult to interact with the web interface. The number of animations and sound effects that they can play with was limited and this might have made them feel that it was easy to control the robot. As future development, more animations and more sound effects will make the robot more expressive. There will be more buttons to push in the web interface.

7.4. Future Work

In this thesis, three systems were built. However, they were not compared with each other. For the future research, a comparison of three different interfaces is required. Quantitative measures as well as qualitative measures are necessary. A task completion time might be a good measure to include in future studies. People find difficulties in finding which buttons to press. If we assume the other interfaces are intuitive enough, they will take less time to evoke different kinds of animations. Both the wearable interface and the sympathetic interface are developed to be more expressive than the web interface. It will also be needed to evaluate the system on the user's side whether such expressiveness actually affect the user on the robot's side or not.

8. Conclusion

The aim of this thesis was to develop general software architecture for a semi-autonomous robot avatar and to develop interface devices to aid puppeteering for human operators.

I developed the overall puppeteering system on top of the existing C5M behavior system architecture and the Microsoft Robotics Developer's Studio. We modified the motor system of the C5M to make the robot look at random places while its parts still being puppeteered by the human user. Thus, the human operator is able to control the gaze of direction and the pointing direction via different interfaces while other parts remain to be autonomous.

We have built the three different user interfaces to control the robot effectively.

The web interface has been improved with the semi-autonomous stale panorama view. It provided enhanced situational awareness to users. All the animations can be smoothly played without any glitches in the robot's movement. Most sub systems have been integrated into the web page and became an all-in-one interface to the human operator. In its human subject study trial, most people felt confident enough to control the interface and finished the task in the given time.

Two proto-types of the wearable interface were built. The gesture recognition interface was built utilizing the Nintendo's Wii controllers. It was able to recognize six different gestures, but the number of recognizable gestures was limited since it only used three axis accelerometer values from the sensor units. We also tested the infrared camera with the wearable markers. For direct

manipulation, it provided an affordable and effective device to control an avatar. However, a human operator had to be concerned about his/her posture all the time to locate the robot parts at the right positions.

The last interface that we built was the sympathetic interface. The semi-autonomous feature of the interface made it possible for a human operator to place the interface on his/her lap and to control the web interface and the sympathetic interface at the same time. It will significantly help lessen the cognitive load of a human operator. He/she does not need to worry whether the interface keeps its posture at a certain position or not.

In conclusion, the interfaces developed in this thesis have great potential to be applied to many different applications. As mentioned earlier, in the future they might be a new media for daily communication where a user can have physical interactions via such interfaces. Specifically, children or elderly people could benefit from education and health care applications using such interfaces. For example, children could learn a second language from the robot while teachers are in remote places. Native speakers can stay in their countries and still offer conversational practice or play in other countries. It might also be possible for elderly people to physically interact with their grand children both through the wearable and sympathetic interfaces.

References

- Benbasat, A. Y., & Paradiso, J. A. (2002). An Inertial Measurement Framework for Gesture Recognition and Applications. *Gesture and Sign Languages in Human-Computer Interaction: International Gesture Workshop, GW 2001, London, UK, April 18-20, 2001: Revised Papers*.
- Blumberg, B., Downie, M., Ivanov, Y., Berlin, M., Johnson, M. P., & Tomlinson, B. (2002). Integrated learning for interactive synthetic characters. *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 417-426.
- Bobick, A. F., Intille, S. S., Davis, J. W., Baird, F., Pinhanez, C. S., Campbell, L. W., et al. (2000). The KidsRoom. *Communications of the ACM*, 43(3), 60-61.
- Bonabeau, E., Dorigo, M., Theraulaz, G., & NetLibrary, I. (1999). *Swarm Intelligence from Natural to Artificial Systems*. Oxford University Press.
- Breazeal, C. (2003). Toward sociable robots. *Robotics and Autonomous Systems*, 42(3-4), 167-175.
- Bregler, C. (1997). Learning and recognizing human dynamics in video sequences. *IEEE Conference on Computer Vision and Pattern Recognition*, 568-574.
- Cassell, J. (2000). Nudge Nudge Wink Wink: Elements of Face-to-Face Conversation for Embodied Conversational Agents. In *Embodied Conversational Agents*. MIT Press.
- CereProc Text to Speech - Homepage. . Retrieved September 27, 2008, from <http://www.cereproc.com/>.
- Dautenhahn, K., & Billard, A. (1999). Bringing up robots or-the psychology of socially intelligent robots: from theory to implementation. *Proceedings of the third annual conference on Autonomous Agents*, 366-367.
- Davis, J. W., & Bobick, A. (1997). The representation and recognition of action using temporal templates. *IEEE Conference on Computer Vision and Pattern Recognition*, 928-934.
- Dementhon, & Davis. (1995). Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1), 123-141. doi: 10.1007/BF01450852.
- Downie, M. (2000). *Behavior, Animation, Music: The Music and Movement of Synthetic Characters*. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA.

- Esposito, C., Paley, W. B., & Ong, J. (1995). Of mice and monkeys: a specialized input device for virtual body animation. In *Proceedings of the 1995 symposium on Interactive 3D graphics* (pp. 109-ff.). Monterey, California, United States: ACM. doi: 10.1145/199404.199424.
- Fong, T., Nourbakhsh, I., & Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42(3-4), 143-166.
- Fontaine, D., David, D., & Caritu, Y. Sourceless human body motion capture. In *Smart Objects Conference (SOC 2003)*.
- Goza, S. M., Ambrose, R. O., Diftler, M. A., & Spain, I. M. (2004). Telepresence control of the NASA/DARPA robonaut on a mobility platform. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 623-629). Vienna, Austria: ACM.
- Gray, J. V. (2004). *Goal and Action Inference for Helpful Robots Using Self as Simulator*. Master's Thesis, Massachusetts Institute of Technology.
- Hancher, M. D. (2003). *A Motor Control Framework for Many-Axis Interactive Robots*. M.Eng. Thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science.
- Hidden Markov Model Toolbox for Matlab. . (2005). Retrieved from <http://cs.ubc.ca/~murphyk/Software/HMM/hmm.html>.
- Ishii, H., & Ullmer, B. (1997). Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 234-241). Atlanta, Georgia, United States: ACM. doi: 10.1145/258549.258715.
- Johnson, M. P., Wilson, A., Blumberg, B., Kline, C., & Bobick, A. (1999). Sympathetic interfaces: using a plush toy to direct synthetic characters. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit* (pp. 152-158). Pittsburgh, Pennsylvania, United States: ACM. doi: 10.1145/302979.303028.
- Maes, P., Darrell, T., Blumberg, B., & Pentland, A. (1997). The ALIVE system: wireless, full-body interaction with autonomous agents. *Multimedia Systems*, 5(2), 105-112.
- Mazalek, A., & Nitsche, M. (2007). Tangible interfaces for real-time 3D virtual environments. In *Proceedings of the international conference on Advances in computer entertainment technology* (pp. 155-162). Salzburg, Austria: ACM. doi: 10.1145/1255047.1255080.
- Microsoft Robotics. . Retrieved September 24, 2008, from <http://msdn.microsoft.com/en-us/robotics/default.aspx>.
- Moeslund, T. B., Hilton, A., & Krÿger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3), 90-126.

- Moeslund, T. B., & Granum, E. (2001). A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding*, 81(3), 231-268.
- Raab, F. H., Blood, E. B., Steiner, T. O., & Jones, H. R. (1979). Magnetic Position and Orientation Tracking System. *Aerospace and Electronic Systems, IEEE Transactions on*, 709-718.
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257-286.
- Sakamoto, D., Kanda, T., Ono, T., Ishiguro, H., & Hagita, N. (2007). Android as a telecommunication medium with a human-like presence. In *Proceeding of the ACM/IEEE international conference on Human-robot interaction* (pp. 193-200). Arlington, Virginia, USA: ACM.
- Schlenzig, J., Hunter, E., & Jain, R. (1994). Recursive identification of gesture inputs using hidden Markov models. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on* (pp. 187-194).
- Shin, H. J., Lee, J., Shin, S. Y., & Gleicher, M. (2001). Computer puppetry: An importance-based approach. *ACM Trans. Graph*, 20(2), 67-94.
- Stan Winston Studio. . Retrieved September 29, 2008, from <http://www.stanwinstonstudio.com/home.html>.
- Starner, T. (1995). *Visual recognition of American sign language using hidden Markov models*. Master's Thesis, Massachusetts Institute of Technology.
- Stiehl, W. D., Lieberman, J., Breazeal, C., Basel, L., Lalla, L., & Wolf, M. (2005). Design of a therapeutic robotic companion for relational, affective touch. *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, 408-415.
- Sturman, D., & Sturman, D. (1998). Computer puppetry. *Computer Graphics and Applications, IEEE*, 18(1), 38-45.
- Tartaro, A., & Cassell, J. Authorable Virtual Peers for Autism Spectrum Disorders. *Combined Workshop on Language-Enabled Educational Technology and Development and Evaluation for Robust Spoken Dialogue Systems at the 17th European Conference on Artificial Intelligence (ECAI06), (Riva del Garda, Italy, 2006)*.
- Telepresence - Wikipedia, the free encyclopedia. . Retrieved December 30, 2007, from <http://en.wikipedia.org/wiki/Telepresence>.
- Tomasi, C., & Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2), 137-154.

- Toscano, R. L. (2008). *Building a Semi-Autonomous Sociable Robot Platform for Robust Interpersonal Telecommunication*. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA.
- Ullman, S., & Basri, R. (1991). Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10), 992-1006.
- Weller, M. P., Do, E. Y., & Gross, M. D. (2008). Posey: instrumenting a poseable hub and strut construction toy. In *Proceedings of the 2nd international conference on Tangible and embedded interaction* (pp. 39-46). Bonn, Germany: ACM. doi: 10.1145/1347390.1347402.
- Wren, C. R., Azarbajejani, A., Darrell, T., & Pentland, A. P. (1996). *Pfinder: real-time tracking of the human body*. Master's Thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science.
- Wren, C. R., & Pentland, A. P. (1999). Understanding purposeful human motion. *Modelling People, 1999. Proceedings. IEEE International Workshop on*, 19-25.
- Yamato, J., Ohya, J., & Ishii, K. (1992). Recognizing human action in time-sequential images using hiddenMarkov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on* (pp. 379-385).
- Yeoh, W., Wu, J., Pek, I., Yong, Y., Chen, X., & Waluyo, A. (2008). Real-time tracking of flexionangle by using wearable accelerometer sensors. *5th International Workshop on Wearable and Implantable Body Sensor Networks*, 125-128.
- Young, A. D., Ling, M. J., & Arvind, D. K. (2007). Orient-2: a realtime wireless posture tracking system using local orientation estimation. In *Proceedings of the 4th workshop on Embedded networked sensors* (pp. 53-57). ACM Press New York, NY, USA.