2007-09-26

Exploring the Use of Tangible User Interfaces for Human-Robot Interaction: A Comparative Study

Guo, Cheng; Sharlin, Ehud

http://hdl.handle.net/1880/45640
unknown

Exploring the Use of Tangible User Interfaces for Human-Robot Interaction: A Comparative Study

Cheng Guo and Ehud Sharlin

ABSTRACT

In this paper we suggest the use of tangible user interfaces (TUIs) for human-robot interaction (HRI) applications. We discuss the potential benefits of this approach while focusing on low-level of autonomy tasks. We present an experimental robotic interaction testbed we implemented to support our investigation. We used the testbed to explore two HRI-related task-sets: robotic navigation control and robotic posture control. We discuss the implementation of these two task-sets using an AIBO robot dog. Both tasks were also mapped to two different robotic control interfaces: keypad interface which resembles the interaction approach common in HRI, and a gesture input mechanism based on Nintendo Wiimotes and Nunchuks. We discuss the interfaces implementation and conclude with a detailed user study we performed to compare these different HRI techniques in the two robotic tasks-sets.

Author Keywords: Human-Robot Interaction, Tangible User Interface, Gesture Input

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces – Interaction Styles

INTRODUCTION

Over the last few decades, a large variety of robots have been introduced to numerous applications, tasks and markets. They range, for example, from robotic arms [4] that are used in space station assemblies to explosive ordnance disposal robots [11] dispatched on battlefields. Depending on the task difficulty and the complexity, the interaction techniques used by the human operator to control a robot may vary from joystick movements, simple mouse clicks or a set of keystrokes on a control keyboard.

When performing tasks the human operator may need to break down high-level commands such as “pick up that object” into a sequence of low-level discrete actions that the robot can perform, and then translate each action to a key or switch on the user interface to trigger the appropriate action on the robotic platform. The necessity to perform high-level robotic actions through the composition low-level actions is not ideal. Depending on the low-level set of interactions the overall experience can be unnatural, error prone and can impose a need for advanced user training. We, like others, are exploring more clear and intuitive spatial mappings between interface and robot to facilitate intuitive, high-level robotic interfaces.

As the level of task difficulty increases, it is ideal if the operator spends more time on high-level problem solving and task planning than on low-level robot operations. Intuitive interfaces would allow users to focus on the content of their human-robot interaction (HRI) tasks rather than on the micro-scale operations needed to accomplish these tasks. Generic input devices such as keyboards and joysticks can hinder higher-level interactive tasks as their physicality (layout of keys and buttons) is limited and cannot always be mapped intuitively to a large set of robotic actions.

The aforementioned problem can be tackled by searching for natural and intuitive input methods for robotic interfaces, with one possible avenue being the use of gestures. Studies have shown that children begin to gesture at around 10 months of age [18] and that humans continue to develop their gesturing skills from childhood to adolescence [17]. This natural skill coupled with speech enables us to interact and communicate with each other more effectively. In contrast, keyboards and mice, which are arguably not difficult to learn, are acquired skills that are not as innate as performing gestures with our hands and arms. Also, the generic nature of the mouse and keyboard cause them to be inappropriate for certain tasks, which can break the flow of users’ cognitive engagement with the task, negatively impacting performance [6]. Can specialized gesture controlled input devices offer more efficient mappings from human to robot than the prevalent keyboard, joystick and mouse interface for a certain set of HRI tasks?

Intuitive, efficient spatial mappings underlie the design of tangible user interfaces (TUIs). TUIs couple digital information and function with physical objects [12] allowing users to interact with a physical object in order to manipulate a virtual entity in the digital realm. TUIs make effective use of the affordances [20] of physical objects which can directly represent their functionality. The shape, size and weight along with other physical properties of a physical object imply the way we interact with it. By taking the advantage of the affordances of physical objects we may design a set of physical robotic interfaces directly mapped to the physical aspects and the potential functionalities of robotic platforms. Furthermore, the spatial orientation and the position of a physical object in relation to its surroundings can reveal additional information and provide interaction insight and task awareness to the manipulator. When controlling a robot, maintaining good human-robot HRI awareness [5] is crucial to the operator. If a physical object can be translated into a tool for controlling robot, then the orientation and position of the object in the physical space can be utilized to provide additional information about the status of a robot. We see great potential for the user of TUIs in HRI and explore in this paper a TUI-based mediator for merging human gestures with robotic actions.

To explore the possibilities of applying TUIs to human-robot interaction, we utilized the Nintendo Wiimote and Nunchuk [19] as a generic TUI for capturing human postures. We mapped these simple TUIs to a set of robotic actions in order to support a robotic interface based on dynamic capture of human postures and gestures. (Figure 1) In order to assess the quality of the Wiimote and Nunchuk interface, we designed an experimental testbed which allowed us to test it against a generic input device – a keypad, with a robot that has 0% autonomy and 100% intervention ratio [34]. Our experimental testbed is based on an AIBO robot dog which the user had to control through a variety of tasks. A user study was then conducted in an attempt to investigate the advantages and drawbacks of each interaction method in practical HRI tasks.

Figure 1. Wiimote, Nunchuk and AIBO

In this paper we briefly present related TUIs efforts and other instances of using gestures in the field of HRI. We describe in detail our Wiimote and Nunchuk interaction technique implementation, the baseline keypad interface and the robotic testbed. We present our experimental design, the comparative user study and its results. We discuss the findings and their implications on using gestures in Human-Robot Interaction tasks vis-à-vis a more orthodox keyboard-based approach.

RELATED WORK

Human-robot interaction (HRI) is a relatively new sub area of study in HCI. A large amount of effort in the field of robotics has been spent on the development of hardware and software to extend the functionality and intelligence of existing robotic platforms. Compare to the substantial increase in the variety of robots and their capabilities, the techniques people use to command and control robots remain relatively unchanged. As robots are being deployed in more demanding situations, the intuitiveness of interaction techniques for controlling robots has raised a considerable amount of attention among HRI researchers. In terms of interface design, Goodrich and Olsen's [9] work provided a general guide to HRI researchers on how to design an effective user interface. Drury, Yanco and Scholtz have defined a set of HRI taxonomies [34] and conducted a thorough study [33] on how to improve human operators' awareness of rescue robots and their surroundings. To broaden the view of HRI researchers in interface design, Richer and Drury [23] had summarized and formed a video game-based framework that can be used to characterize and analyze robotic interfaces.

Yanco and Drury defined and detailed sets of robotic interfaces terminologies and definitions in an effort to classify the different HRI approaches explored in the domain [34]. Specially, they defined that a robot's autonomy level is being measured as the percentage of time that the robot is carrying out its task on its own. In correspondence, the amount of intervention required for a robot to function is measured as the percentage of time that a human operator must be controlling the robot. These two measures, autonomy and intervention, sum up to 100% [34]. We use this definition in our work, mainly focusing on robotic interfaces with low autonomy levels.

Many current efforts are being conducted by HRI researchers in an attempt to broaden the capabilities and improve the quality of communication between humans and robots. We are inspired by these efforts, and in this work we try to address some of the fundamental HRI challenges in the still common low autonomy level robotic tasks where a high level of human intervention is required.

The notion of tangible user interfaces [12] is based on Fitzmaurice's earlier Graspable User Interfaces effort [8]. Fitzmaurice and Buxton have conducted an experiment which allowed users to use "Bricks" [7] as physical handles to direct manipulate virtual objects. Their study has shown that *a space-multiplex input scheme with specialized devices can outperform a time-multiplex (e.g., mouse-based) input design for certain situations*. [7] Later, Ishii and Ullmer proposed the term Tangible User Interfaces (TUIs) in TangibleBits [12] paper. Ishii and Ullmer's research addressed the importance of both the foreground interaction which consists of using physical objects to manipulate virtual entities and the background interaction which happens at the periphery to enhance users' awareness using ambient media in an augmented space. In our research, we focus on the essence of TUIs which is defined by Ishii as "*seamless coupling everyday graspable objects with the digital information that pertains to them*" [12]. Moreover, we want to select a TUI that has a tight spatial mapping [24] with robots. Spatial mapping can be defined as "*the relationship between the object's spatial characteristics and the way it is being used*" [24]. A "good" TUI for HRI should take advantage of its physical and spatial characteristics to reflect the physical state or function of robots.

Another quality of TUIs that can make them an interesting choice for HRI tasks is I/O unification, or the natural coupling of action and perception space [24]. TUIs, like any physical object, can allow us to perceive and act at the same place and at the same time. This quality is often lost in orthodox interfaces that usually separate action space (e.g. mouse) from perception space (e.g. display). By capturing this natural quality of physical objects TUIs can support the implementation of robotic interfaces that are more intuitive and efficient, allowing the user to be more attentive and focus on the task at hand.

Using gestures to interact with robots is not a new idea in the field of HRI. A significant amount of work has been done using either vision based or glove based systems to capture human arm and hand gestures. Among these efforts, we found that Korenkamp et al.'s [14] work is somewhat similar in approach to the gesture recognition technique we used. Korenkamp et al. presented a vision-based technique to monitor the angles between a person's forearm and upper arm to predict the gesture that the person is performing. For our approach, we used the Wiimote and Nunchuk to detect the rotation angle of a person's shoulder and elbow joints in relation to the arm rest position.

Another interesting approach to use the human body as an input device to interact with Robots is the exoskeletons system [3, 13]. Kazerooni's research was focused on augmenting the human body with robotic arms and suits to extend the physical strength of an individual. In his research, the human operator wore a robotic arm to directly apply *mechanical power and information signals* [13] to the robot. By measuring the dynamic contact force applied by the human operator, the robotic limbs are able to amplify that force for performing heavy duty tasks that normal human strength would not be able to afford to. The Robotnaut project [3] uses similar concepts but a different approach to interact with robots. Bluethmann et al. adopted a master-slave system approach which requires the human operator to wear gloves equipped with Polhemus trackers for detecting arm and hand positions [3]. The Robonaut operator does not physically contact with the robot in any way. Thus, there is no mechanical power exchange in between them. This approach allows the human operator to control the Robonaut from a remote distance. Both of the interaction techniques mentioned above allow human operators to directly manipulate a robot that is either collocated or remotely located.

SYSTEM DESIGN AND IMPLEMENTATION

In order to explore the possibility of using gestures for HRI, we were looking for a robotic platform that would allow us full and flexible control in lab settings. The robot should be able to response to both high level commands (such as walking or turning) and to low level commands (such as rotate a specific joint by a certain number of degrees) to match the meaning of both abstract gestures (such as arbitrary hand gestures used in a speech) and specific gestures (such as teaching others a specific movement by demonstrating a similar gesture). Moreover, we were searching for an anthropomorphic or zoomorphic robot that resembles the human skeletal structure in some way in order to achieve an intuitive mapping between the user interface and the robot. In search for the robots that satisfy the above criteria, we found that the Sony AIBO robot dog can be a suitable platform for our experiment. The AIBO is a zoomorphic robot that resembles parts of the human skeletal structure. For instance, the AIBO has "shoulder" and "elbow" joints on its forelegs which act similarly to human's shoulder and elbow joints. The AIBO is fully programmable through the use of the Tekkotsu framework [27]. Developers can gain control over the low level sensor and actuators and high level body gestures and movements.

To evaluate the usability of gesture input for HRI in contrast with a generic input device, we have designed two interaction techniques for manipulating an AIBO robot dog in a collocated setup. One of the interaction techniques supports human gesture input through a Wiimote and Nunchuk interfaces, another input technique uses a keypad as the basis for interacting with an AIBO. During the selection of TUIs for our research, the Nintendo Wiimote comes to our attention. The Wiimote clearly differentiates itself from other generic controllers in terms of the interaction style. Instead of pressing buttons, the Wiimote allows players to use motions such as, swing, shake and thrust to interact with the virtual objects on the TV screen. Players feel more involved and satisfied when using the Wiimote due to the fact that virtual entities in games response and react to physical input. Although the Wiimote does not appear to be a specialized TUI as the ones presented by Ishii and Ullmer, we believe it can be categorized as a generic tangible user interface due to its ability to capture physical input and to interact with digital entities. In order to utilize the power of Wiimote and apply it to control an AIBO, we used a PC equipped with both Bluetooth and 802.11b wireless network adapter to act as a mediator to translate and transmit the command from the Wiimote to the AIBO.

Another interface that we selected for representing the generic input device is an OQO 02 Ultra-Mobile PC [21] (UMPC) with an onboard thumb keyboard. The OQO 02 is a scaled down version of a regular desktop PC. It has built-in wireless network adapter that can be used to communicate with an AIBO. We believe this "button-press and key-action mapping" interaction style represents a common interaction technique in HRI today.

One important thing to note for the experiment is that there was a small amount (about half a second) of action delay on the AIBO side after the participants send a command to it. This lagging issue is unavoidable since the controlling program has to sense the user input and then transmit it to the AIBO through a 802.11b wireless network. This transition takes time. To maintain the fairness of the experiment, the underlying controlling code for both techniques is identical. Thus, the amount of lag the participants experienced should be the same with both techniques unless there were random spikes occurred on the wireless network.

Wiimote & Nunchuk Input

The Wiimote consists of a D-pad, a speaker, four LEDs and eight buttons on itself. It communicates with the Wii via Bluetooth wireless link. A complete 3-axis accelerometer [1] measures a minimum full-scale range of ± 3g with 10% sensitivity is located inside the Wiimote. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration. [1] The build-in PixArt optical sensor allows the Wiimote to determine where it is pointing at on a screen with the presence of a sensor bar [19]. The Wiimote also contains a motor with an unbalanced weight inside to support vibration. [30] An extension port is located on the bottom of a Wiimote to allow peripherals such as, a Nunchuk to be attached. The Wiimote is powered by a pair of AA batteries.

To extend the functionality of a Wiimote, a Nunchuk can be connected to a Wiimote via a cord. The Nunchuk has an analog stick and two buttons on itself. It uses the same accelerometer on the Wiimote to support motion sensing.

Accelerometer

In order to understand the Wiimote's motion sensing capability, we need to exam its acceleration measuring mechanism first. According to the Data sheet [1] of the ADXL 330 accelerometer:

"The sensor is a polysilicon surface micromachined structure built on top of a silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and plates attached to the moving mass… Acceleration deflects the moving mass and unbalances the differential capacitor resulting in a senior output whose amplitude is proportional to acceleration."

In other words, the sensor does not measure the acceleration of the Wiimote, but rather the force exerted by the test mass on its supporting springs. [31] When the Wiimote is at rest on a flat surface, the accelerometer reading is 1 g (approximately 9.8 m/s^2) due to gravity. When it is in a free fall motion, the reading is close to zero. These facts implies: one, we can only derive a relatively accurate measuring of the pitch and roll angle of the Wiimote when it is reasonably still. This is because, when the Wiimote is accelerating (e.g. a user is swinging the Wiimote), the acceleration value sensed by the Wiimote is due to the force exerted by the user rather than the pulling of gravity. Thus, the tilting angle derived based on this force does not represent the current position of the Wiimote. Two, the accelerometer cannot detect the rotation angle around the gravitational axis. For instance, when the Wiimote is facing up (e.g. the A button is facing upward) and rest on a flat surface, the Z-axis (Figure 2) of the accelerometer is parallel to the direction of gravity. Thus, it does not matter how we orient the Wiimote on the surface, the acceleration value sensed on the Z-axis always remain the same. This means we lose one degree of freedom when one of the axes of the accelerometer aligns with the direction of the gravity.

Due to the constraints associated with the accelerometer and the unavailability of a motion analyzing package, we are left with the choice of measuring pitch and roll for recognize arm and hand gestures. In our experiment, we want to allow people to use large arm movements for controlling an AIBO, because large movements are easier to distinguish thus easier to memorize. Therefore, we decide to rely on only using the pitch angle of the Wiimote and Nunchuk to predict arm positions. In this case, we use the Wiimote and Nunchuk as a one degree of freedom input devices to measure the rotation angle of a person's elbow and shoulder joint in relation to the arm rest position.

Figure 2. The coordinate system of e Wiimote

In order to access the acceleration value sensed by the Wiimote and Nunchuk, we used Brian Peek's C# library [16] for acquiring the accelerometer readings. To covert the raw acceleration value into rotation angles, we enter the calibrated raw acceleration values into the formula [31]: Pitch = asin (a_y / 1). The variable a_y denotes the calibrated acceleration value along the Y-axis.

OQO 02 Thumb Keyboard

The generic input device that we used is the thumb keyboard on an OQO 02 model. The OQO 02 is equipped with a 1.5 GHz VIA C7M ULV CPU, 1GB DDR2 SDRAM and a 60GB HDD. It runs on Windows® XP Tablet PC Edition 2005. It supports both 802.11 a/b/g and Bluetooth network standards. The input devices on the OQO 02 include a dedicated mouse, a backlit thumb keyboard with a total of 58 keys (including function keys, letter keys and a number pad) and a digital pen. The letter keys on the thumb keyboard follow the QWERTY keyboard layout. The OQO 02 can be either powered by a removable lithium-ion polymer battery or an AC charger. For our comparative study, we used the thumb keyboard only for controlling an AIBO.

The Tekkotsu Framework

We used the Tekkotsu Framework as a mediator to access to the sensors and actuators on an AIBO. The Tekkotsu framework was developed by Carnegie Mellon University. It provides a high level API for programmers to directly control motor level behaviors of an AIBO. It also allows programmers to create complex behaviors based on simple gestures and store it on a memory stick that is inserted into the AIBO. The Tekkotsu Framework allows programmers to remotely access the behaviors that run on an AIBO from any wireless network enabled computing devices.

EXPERIMENTAL DESIGN

To compare and better understand how well people can learn and utilize the aforementioned techniques when controlling a robot we designed an experimental testbed based on two tasks comparing the techniques in terms of speed, accuracy and subjective preferences of the participants. Our goal was to explore the benefits and drawbacks associated with each interaction technique and to try to point out which technique supports a more effective, intuitive and rich user experience when interacting with a robot.

Pilot Study

Before the user study, we had conducted a pilot study to test the usability of both interaction techniques and the fairness of both techniques under different conditions. We found that our posture recognition technique does not suit well with people who have large body size. Thus, we changed our system to allow for a more flexible range of input. However, misrecognition still occurred during the pilot study. To minimize the impact of this problem on the participants' task completion time, we modified the underlying software component that supports the interaction to automatically record the time when each posture command is triggered. The examiner also used the same software to manually log the time when a correct posture is preformed by pressing a button on a keyboard. Moreover, a video tape recorder is used to capture the entire experiment for replay and time synchronization purpose.

To enable participants to navigate the AIBO, we initially used the "W, A, S and D" key mapping on the OQO keypad for the navigation test. However, due to this particular key arrangement, users only need to use their left thumb for most of the movements they need to perform. On the other hand, with the Wiimote technique, users have to use both hands with equal amount of effort to navigate the AIBO. It is not fair to compare a single hand interaction technique with an asymmetric bimanual [2] interaction technique. Thus, we changed the key mapping for the keypad interface (which is explained in detail in the user study navigation task section) to preserve the fairness of the comparison study.

USER STUDY

Participants

For the comparative user study, we recruited twenty participants (16 males and 4 females) from the University of Calgary. Ages ranged from 18 to 29 (M = 21.75, SD = 3.05). All of the participants reported to use computer keyboard everyday. Among all of the participants, eighteen people were right-handed, one person was left-handed and one person was ambidextrous. All of the participants indicated that they have some sort of computer game experience. Fifteen participants reported to play computer games on a weekly and daily basis. Seventeen participants indicated that they "often" or "very often" use computer keyboard to play games. Six participants reported no prior experience playing the Nintendo Wii. Out of the fourteen people who had previous experience with the Wii only three participants reported to play it on a weekly basis. The other 11 indicated playing either "Monthly" or "Rarely".

Task and Procedure

Our experiment was designed for two different tasks, *robotic navigation* and *robotic posture*, each with two difficulty levels. The participants were asked to perform both tasks with both interaction techniques. Thus, in total, participants had to go through four sub experiments in order to complete the study. The order of techniques was counterbalanced among participants by alternating the tasks order, thus ten participants started with the Wiimote Interface and ten participants started with the OQO interface. The experiment was conducted following a written protocol that was read out loud to the participant, starting with an introduction to the experimental testbed and its purpose. Participants were asked to start with one interaction technique to complete both *navigation* and *posture* tasks and then switch to another technique and repeat the two tasks. During the experiment, each participant was asked to complete four questionnaires for each interaction technique for each task. After the experiment, each participant was asked to complete a post-study questionnaire which was followed up with a non-structured interview.

To allow participants to learn and practice each interaction technique and to familiarize themselves with the tasks a practice trail was administrated before the full experiment started. The administrator demonstrated the interaction techniques and present guidelines on how to complete the tasks. Then, the participants would try out the interaction technique and experience it by themselves until they felt proficient and comfortable in handling the robot through the interface and were ready to move to the full experiments.

The main dependent measure in the experiment was the task completion time. In addition, we recorded and analyzed the number of errors that the participants made during the experiment for each task with each interaction technique.

Task 1 – Navigation

Description

In this task, the participant is asked to navigate the AIBO through an obstacle course (Figure 3). The obstacle course is 262 cm in length and 15.3 cm in width. The goal of this test is to see how well both interaction techniques support user control in a fairly delicate robotic navigation task. We provided the user with eight different navigation control actions: *walk forward, stop, walk forward while turning left, walk forward while turning right, rotate left, rotate right, strafe left and strafe right*. To motivate the participants to use all actions, we used two difficulty levels in our trails. For the easier trial participants were not forced to use any particular actions during the course of the obstacle trail and could have chosen any combination of actions they want.

However, for the harder trial, participants were forced to use rotation and strafing in addition to walk and turning in order to complete the obstacle course successfully. A dotted yellow line on the course (Figure 3) indicated the starting point of the strafing action. The solid yellow line indicated the starting point of the *rotate right* action. In order to finish this task, the participants were asked to complete the easier trail first followed by the harder trail.

Before the start of the experiment, we reminded the participants to complete the obstacle trail as fast as possible, and try to make as few errors as possible. An error in this task was defined as hitting obstacles, navigating the AIBO out of the trail boundary or failure to perform the required action at certain locations. If a participant navigated the AIBO out of the trail boundary, then she/he had to navigate it back to the trail and continue on. If a participant failed to perform the required action at certain locations during the trail the examiner had to physically move the AIBO back to that location and ask the participant to try again. This error correction mechanism could have introduced a variable amount of time into the task completion time depending on how fast the examiner moves the AIBO back to the right location. We emphasized the "penalizing" implications of this set of errors to participants and were pleasantly surprised to see that none of the experimental trails required the administrator to physically move the AIBO or to manually correct any out-of-bound navigation errors.

The robotic navigation task is a high level effort which required the participants to engage in route planning while navigating the AIBO. The participants did not have to consider low-level control of the four AIBO legs while controlling actions such as *move forward* or *strafe*. But instead, users had to focus on selecting and performing the appropriate action at the right location and the right timing to complete the obstacle course effectively.

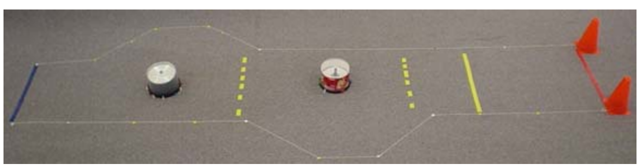

Figure 3. The obstacle course

Interaction Techniques

For this task, the function mapping for the Wiimote interface is presented in Figure 4 and the mapping for the keypad interface is presented in Figure 5:

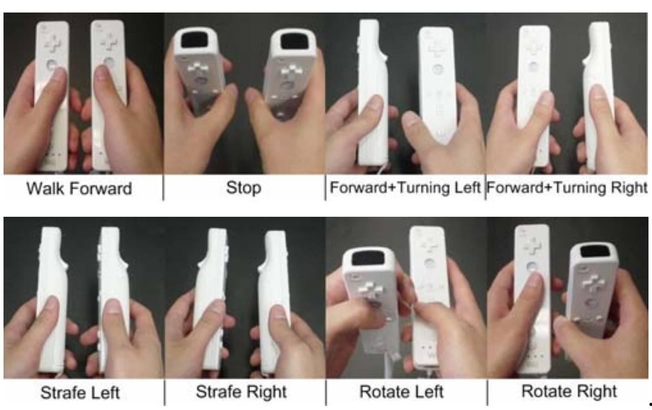

Figure 4. The Wiimote interaction technique for controlling the movement of the AIBO

Figure 5. Key – Movement Mapping

The key-movement mappings are: Forward – W + 2, Stop – S + 5, Forward + Turning Left – A + 2, Forward + Turning right – W + 6, Strafe Left – A + 4, Strafe Right – D + 6, Rotate Left – S + 2, and Rotate Right – W + 5. The plus sign means by pressing and holding the keys on both side of the sign.

The Data collected from this task was analyzed using a 2 x 2 within-subjects ANOVA for the following factors:

- *Technique*: Wiimote, Keypad
- *Difficulty*: easy, hard.

Task 2 - Posture

Description

This task is used to examine the usability of both interaction techniques at low level robot control. In this task, we asked the user to perform twelve different postures with the forelegs of the AIBO. We display an image of the AIBO with a posture on a computer screen. Then the participants control the AIBO to imitate that posture. In the experiment setup, we have pre-defined four different postures for each foreleg of the AIBO. (Figure 6) We selected ten postures out of the sixteen possible combined postures using both forelegs. Then, we divided them into two groups. (Figure 11) Each posture within its own group can be chained together to form a complete gesture. The only difference between these groups of postures is that in order to transform from one posture to another within a group, the participants have to manipulate either one foreleg or both forelegs of the AIBO to complete the transition. We define the group of postures that require only one arm movement during transition as the easier set, and the other group as the harder set. For the experiment, the participants were asked to perform the easier set first followed by the harder set.

Similar to task 1, we measure the task complete time and number of errors. The task complete time in this task is defined as the time elapsed since a new posture image is displayed on the screen till the time the participants invoke (a button press or perform the posture) the correct posture. The error in this case is defined as performing a posture that is different from the posture displayed on the screen. If a participant fails to perform the correct posture, then he/she needs to correct themselves. The time it takes the participants to think and correct their posture is also taken into account of the task completion time. Since the harder posture set requires the participants to send two commands to control both forelegs of the AIBO, these commands can be sent either simultaneously or sequentially. In this case, we did not constrain the participants to any of the input styles. They may choose any input style as long as they feel it is the fastest and most intuitive way to complete the postures.

This task is a low level task which requires the participants to directly adjust the "shoulder" and "elbow" joints of the AIBO to perform certain postures. The participants do not need to worry about task planning and other high level managements mentally. They only need to distinguish the posture image displayed on a monitor and focus on the controlling interface to manipulate the AIBO to imitate it.

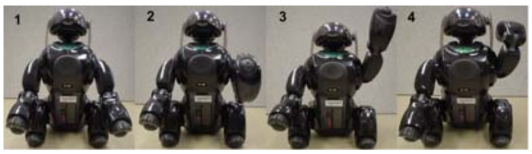

Figure 6. The possible postures for each foreleg of the AIBO

Interaction Techniques

For this task, the function mapping for the Wiimote interface is presented in Figure 7 and the mapping for the keypad interface is presented in Figure 8:

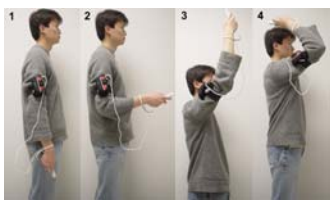

Figure 7. Arm posture input. These postures correspond to the four AIBO postures showed in Figure 7.

Figure 8. Key-Posture Mapping.

The four letter keys control the right foreleg of the AIBO. The four number keys control the left foreleg of the AIBO. By pressing either X or 8, the AIBO will perform Posture 1 (Figure 6) with either it right foreleg or left foreleg. By pressing either Z or 9, the AIBO will perform Posture 2. By pressing either A or 6, the AIBO will perform Posture 3. By pressing either Q or 3, the AIBO will perform Posture 4 (Figure 6).

The Data collected from this task was analyzed using a 2 x 2 within-subjects ANOVA for the following two factors:

- *Technique*: Wiimote/Nunchuk, Keypad
- *Posture*: posture 1 to 12 (Figure 9).

Figure 9. Posture 1-6 is the easier posture group. Posture 7-12 is the harder posture group.

RESULTS

Task 1 - Navigation

Task Completion Time

A 2 x 2 (*Technique* X *Difficulty*) ANOVA, with repeated measures on both factors, revealed no significant *Technique* X *Difficulty* interaction ($F_{1,19}$ = 1.54, p = 0.23), which suggests that performance with the techniques is not substantially influenced by the difficulty level. There was a significant main effect for *Technique*, $F_{1,19}$ = 12.19, p <.001, indicating that overall task completion time for the Wiimote technique (M = 43.2s, SD = 6.9s) was 10% faster than for the keypad technique (M = 48.5 s SD = 6.7s). (Figure 10)

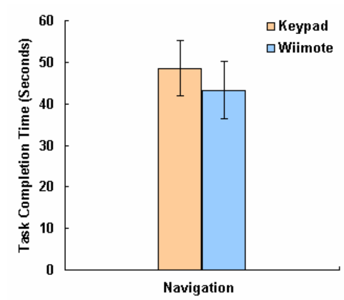

Figure 10. Mean Task Completion Time for the Navigation Task.

As we expected, the main effect of *Difficulty* was significant, $F_{1,19}$ = 115.61, p < .001, with the mean jumping from M = 38.7s, SD = 4.6 s for the easy trail to M = 53.0 s, SD = 8.1s for the hard trail.

Error

A two-way ANOVA was used to determine if there were differences on the number of errors (dependent variable) participants made using the Wiimote and keypad techniques when performed the navigation task under different difficulty levels. The result of the ANOVA showed no significant *Technique* X *Difficulty* interaction ($F_{1,19}$ = 0.03, p = .87), which suggests that the number of errors made using different techniques is not significantly influenced by the difficulty level. There was a significant main effect for *Technique*, $F_{1,19}$ = 9.81, p < .01, indicating the errors that participants made using the Wiimote technique (M = 0.35, SD = 0.4) is 43% less than using the keypad technique (M = 0.83, SD = 0.6). The result also showed a marginally significant main effect for *Difficulty* ($F_{1,19}$ = 3.96, p = .06), with mean varying from M = 0.43, SD = 0.4 for the easy trail to M = 0.75, SD = 0.6 for the hard trail.

Task 2 – Posture

Task Completion Time

A 2 x 12 (*Technique* X *Posture*) ANOVA on the task completion time for the posture task showed a significant *Technique* X *Posture* interaction effect ($F_{11,209}$ = 8.43, p < .001), which means that the *Technique* effect varies with *Posture* or vice versa.

On the average, there was a significant effect for *Technique* ($F_{1,19}$ = 67.37, p < .001), with mean times reducing from 2.2 s (SD = 0.4 s) with keypad, to 1.5 s (SD = 0.3 s) with Wiimote/Nunchuk; On the average, a 32% reduction in task completion time between the two conditions. On the average, pairwise comparisons showed that there was a significant difference (p < .05) between the techniques for posture 1, 2, 7, 8, 9, and 10. But, there was on significant difference for the other postures. (Figure 11)

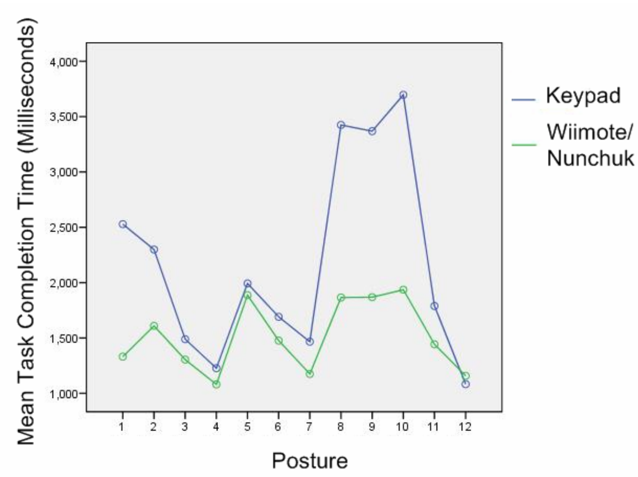

Figure 11. Pairwise comparisons of the mean task completion time for each interaction technique according to posture.

Also, on the average, the test showed a significant effect for *Posture* ($F_{11,209}$ = 27.77, p < .001).

Error

For the keypad interface, participants had made 1.5 (SD = 1.2) errors on average for both difficulty levels. However, none of the participants had made any errors using the Wiimote/Nunchuk interface. As anticipated, a paired t-test showed a significant difference (t_{19} = 7.44, p < .001) between the techniques.

DISCUSSION

The results presented in Section 6 point to the Wiimote and the Wiimote/Nunchuk interfaces outperforming the keypad interface in terms of task completion time in both the *robotic navigation* and the *robotic posture* tasks. The differences between the interfaces, although statically significant, are a little underwhelming in their magnitude. When attempting to explain this for the navigation task we should consider that both interaction techniques use a set of abstract key and gesture combinations to represent specific robot movements. Since none of the participants have prior experience with these input methods before the participants have to learn and memorize the mappings of both techniques in order to navigate the AIBO. This abstract mapping between the user interface and the robot action in the navigation tasks added an extra layer of cognitive load onto the participants to process in both interfaces. Although pressing buttons should not be slower than performing gestures, the study showed that the participants finished the obstacle trails quicker with gesture input than with button input. We believe that although both interfaces require the participants to think about an abstract mapping before they press a button or perform a posture, the Wiimote interface provides a slight advantage. When using the Wiimote participants do not need to focus on their hands when performing a posture, since naturally they are aware of their spatial location. On the other hand, we observed that the participants have to constantly shift their attention back and forth between the keypad and the AIBO to find the buttons they want to press and to confirm if they pressed the right button. The consequences of shifting attention constantly between the interface and the AIBO may result in action overshot (for example, overturning a corner) and can break the continuity of the task when participants have to stop the AIBO before they decide to which action to take for the next step. This practical separation of action and perception spaces is perhaps the reason for the slower task completion time when using the keypad.

Although the study results indicated that gesture input is faster for the navigation task we are not suggesting it would always a better solution than button input for this type of tasks. As we mentioned earlier in section 4 the keypad mapping that we used was arguably not the most intuitive mapping we can come up with. A “W, A, S, D” key configuration would probably be more intuitive to use since it requires less key combinations and is a commonly used mapping in computer games for navigational tasks. However, we believe that our results demonstrate that when participants are limited to use asymmetric two-hand interaction techniques to control a robot, gesture input tends to be more intuitive to use than button input.

For the navigation tasks we did not expect that there would be a significant difference between the numbers of errors participants made using the different techniques. However, the data showed the opposite. Participants made 43% more errors with the keypad interface than with the Wiimote interface in the navigation tasks. Many participants felt that this was due to the small key size and the unintuitive mapping between buttons and robot actions.

For the *robotic posture* tasks, we can see that on average there was a significant difference in task completion time between the postures that required two arms movement and the ones that only required one arm movement. By observation, we found that when the participants were using the Wiimote/Nunchuk interface, they were extremely engaged and focused on the computer screen that displayed the posture images. However, when the participants used the keypad interface, they often looked at the computer screen first, and then focus on the keypad to find the right button to press. This attention shifting problem slowed down the participants' task completion time and can again be associated with the separation between action and perception space created by the keypad. Also, most participants felt they were simply mimicking the postures on the computer screen while using the Wiimote/Nunchuk interface, while they felt the keypad interface required them to "act". Following, we believe that the intuitiveness of gesture input had definitely reduced the cognitive load of associating user inputs with zoomorphic robotic actions.

Moreover, gesture input tends to support simultaneous input compared to button input. As one of the participants commented, "I could do both hands (both arm movements) at the same time without a lot of logical thinking (with the Wiimote/Nunchuk interface), where with the keyboard I had to press one (button) and the other (button) if I was doing two hand movements at the same time. Although they would be in succession, they would not be at the same time."

It is worth to point out that even though posture 1 and 2 only required single arm movements, there was a significant difference between the task completion times of both techniques. In our opinion, we think this is perhaps due to the participants not being fully trained at the beginning of the study. Thus, they tend to make more mistakes with the first few postures. This may also imply that the Wiimote/Nunchuk interface was easier to learn compared to the keypad interface and can be utilized faster.

Subjective Ratings

We also asked the participants to rate the intuitiveness of both input techniques and indicate their preferred techniques for both tasks. Figure 12 and 13 shows the results of participants' ratings.

After the study, we asked the participants who preferred to use the keypad for the navigation task about their subjective reasoning. All users responded that they are more familiar with the keypad interface because of related computer game experiences. However, when it comes to the real experiment, they found the keypad to be harder to use. One of the participants commented, "I have to think harder when I use the keyboard, and this kind of mental overhead coupled with the lag time just makes it feel harder."

For the participants who preferred to use the keypad for the posture task, their reasoning was that they can easily memorize the key-action mapping since there were only four postures for each arm and the buttons associated with both arms are symmetrical on the keypad layout. As one of the participants stated, "With so few postures available, the keyboard was just as easy as the Wiimote." We agree with this participant's comment. We believe that if we provided extensive training to all of the participants using the keypad interface they would eventually probably master this interaction technique and reach comparable, if not outperform, the Wiimote/Nunchuk in terms of task completion time. However, if we increase the number of postures to an amount that participants cannot easily memorize, or if we deal with an interaction task that cannot afford intensive training; we believe that our results demonstrated that the gesture input would be a better interaction choice.

Q1	I found the keypad technique is easy to learn for the Navigation task.
	I found the Wiimote technique is easy to learn for the Navigation task.
Q2	I had difficulty remembering how to perform certain movements with the keypad technique for the Navigation task.
	I had difficulty remembering how to perform certain movements with the Wiimote technique for the Navigation task.
Q3	I found the keypad technique is easy to learn for the Posture task.
	I found the Wiimote technique is easy to learn for the Posture task.
Q4	I had difficulty remembering how to perform certain postures with the keypad technique for the Posture task.
	I had difficulty remembering how to perform certain postures with the Wiimote technique for the Posture task.

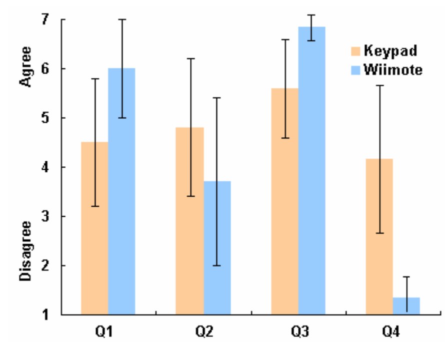

Figure 12. Mean ratings on post-study questionnaire. The rating scale ranges from 1 (strongly disagree) to 7 (strongly agree).

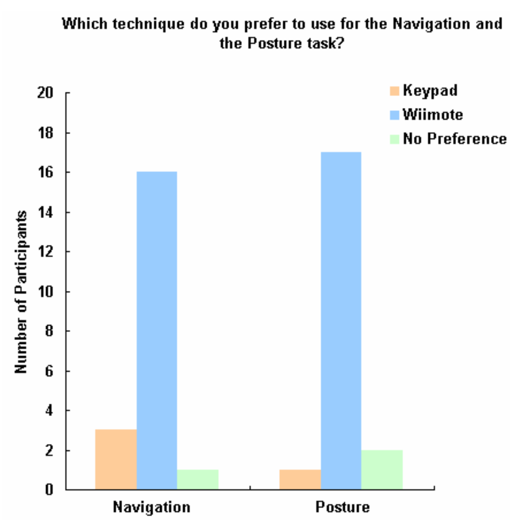

Figure 13. Users' preference for each interaction technique

CONCLUSION

In this paper, we have introduced a new interaction technique which utilizes the Nintendo Wii Remote and Nunchuk to capture human arm and hand gesture input for human-robot interaction. To evaluate this technique, we have conducted a comparative user study which compares the Wiimote/Nunchuk interface with a traditional input device – keypad in terms of speed and accuracy. Two different tasks are deployed to evaluate these two interfaces for both high-level and low-level robot controlling tasks. The result of our experiment provides some evidence that a gesture input scheme with tangible user interfaces can outperform a button-pressing input design for certain situations. We have observed a significant increase in task completion time and decrease in the number of mistakes participants made for both the navigation and posture tasks. From the follow-up questionnaire, we discovered that a significant majority of the study participants chose the Wiimote/Nunchuk interface as their preferred technique for controlling an AIBO in both tasks.

In future work, we hope to improve the Wiimote/Nunchuk interaction technique to analyze continuous human arm and hand gestures to extend our abilities in controlling anthropomorphic and zoomorphic robots. Also, to continue our journey in exploring novel interaction techniques for human-robot interaction, we intend to discover a set of more specialized tangible user interfaces to further blur the barrier between humans and robots.

REFERENCES

1. ADXL330, Analog Devices. http://www.analog.com/UploadedFiles/Data_Sheets/ADXL330.pdf
2. Balakrishnan, R., Hinckley, K. Symmetric bimanual interaction. In *Proc. CHI 2000*, ACM Press (2000), 33-40.
3. Bluethmann, W., Ambrose, R., Diftler, M., Askew, S., Huber, E., Goza, M., Rehnmark, F., Lovchik, C., and Magruder, D. Robonaut: A robot designed to work with humans in space. Autonomous Robots,2003, 14(2-3): 179-197.
4. Canadarm, http://www.nasa.gov/mission_pages/station/structure/elements/mss.html
5. Drury, J. L., Scholtz, J., Yanco, H.A. Awareness in human-robot interactions. In *Proc. IEEE International Conference on Systems, Man and Cybernetics* 2003.
6. Faisal, S., Cairns, P., and Craft, B. Infoviz experience enhancement through mediated interaction. In *Proc. ICMI* 2005 Workshop on Multimodal interaction for the visualization and exploration of scientific Data (2005), 3-9.
7. Fitzmaurice, G.W., Buxton, W. An empirical evaluation of graspable user interfaces: towards specialized, space-multiplexed input. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, ACM Press (1997), 43-50.
8. Fitzmaurice, G.W., Ishii, H., and Buxton, W. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proc*. *CHI* 1995. ACM Press (1995), 317-324.
9. Goodrich, M., and Olsen, D. Seven principles of efficient human robot interaction. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, 2003, 3943–3948.
10. Hasanuzzaman, Md., Zhang, T., Ampornaramveth, V., Kiatisevi, P., Shirai, Y., and Ueno, H. Gesture Based Human-Robot Interaction Using a Frame Based Software Platform. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, IEEE SMC(2004), 2883-2888.
11. iRobot, http://www.irobot.com/sp.cfm?pageid=138
12. Ishii, H. and Ullmer, B. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proc. CHI* 1997, ACM Press (1997), 234-241.
13. Kazerooni, H. Human-Robot Interaction via the Transfer of Power and Information Signals. IEEE Trans. on System and Cybernetics (1990) Vol.20, No. 2, 450-463.
14. Kortenkamp, D., Huber, E., and Bonasso, R. Recognizing and interpreting gestures on a mobile robot. In *Proc. AAAI* 1996.
15. Makinson, B.J. Research and development prototype for machine augmentation of human strength and endurance, Handiman I project. 1971, Rep. S-71-1056, General Electric Schenectady, NY.
16. Managed Library for Nintendo's Wiimote, http://blogs.sdn.com/coding4fun/archive/2007/03/14/1879033.aspx
17. McNeill, D. Hand and Mind: What Gestures Reveal about Thought. University of Chicago Press (1992).
18. Messing, L., Campbell, R.Gesture, speech, and sign. New York: Oxford University Press (1999), 227.
19. Nintendo Wii Controllers , http://wii.nintendo.com/controller.jsp
20. Norman, D.A. 1988. The Psychology of Everyday Things. BasicBooks.
21. OQO 02, http://www.oqo.com/products/index.html
22. Radhakrishnan, S., Potter W.D. Gesture Recognition on AIBO. *IAAI* 2006.
23. Richer, J., Drury J.L. A Video Game-Based Framework for Analyzing Human-Robot Interaction: Characterizing Interface Design in Real-Time Interactive Multimedia Applications. In *Proc. HRI 2006*, ACM Press (2006), 266-273.
24. E. Sharlin, B.A. Watson, Y. Kitamura, F. Kishino, and Y. Itoh, "On Tangible User Interfaces, Humans and Spatiality", Personal and Ubiquitous Computing, Springer-Verlag, 2004, pp 338-346.
25. Sony AIBO Official Site, http://www.sony.jp/products/Consumer/aibo/
26. Sturman, D., Zeltzer, D. A Survey of Glove-Based Input. *IEEE CG&A* (1994), Vol. 14, No. 1, 30-39.
27. Tekkotsu framework, http://www.cs.cmu.edu/~tekkotsu/
28. Weiser, M. *The Computer for the 21st Century*. *Scientific American* (1991), 265(3), 94-104.
29. WiiBrew Wiki, http://wiibrew.org/index.php?title=Wiimote#Accelerometer
30. Wiimote, http://www.wiili.org/index.php/Wiimote#Rumble
31. Wiimote Motion Analysis, http://www.wiili.org/index.php/Motion_analysis
32. Wiimote Motion Sensor, http://www.wiili.org/index.php/Wiimote#Motion_Sensor
33. Yanco, H.A., Drury, J.L. and Scholtz, J. Beyond Usability Evaluation: Analysis of Human-Robot Interaction at a Major Robotics Competition. Journal of Human-Computer Interaction (2004), Volume 19, Numbers 1 and 2, pp. 117-149.
34. Yanco, H.A., Drury, J. Classifying Human-Robot Interaction: An Updated Taxonomy. *IEEE Conference on SMC*, 2004