Direct manipulation of 3-D objects through multimodal control: Towards a robotic assistant for people with physical disabilities

by

Z. Kazi, M. Salganicoff, M. Beitler, S. Chen, D. Chester and R. Foulds

ASEL Technical Report #ROB9509

asel

Applied Science & Engineering Laboratories
University of Delaware/A.I. duPont Institute
1600 Rockland Rd., P.O. Box 269
Wilmington, DE 19803
Phone: (302)651-6830 FAX: (302)651-6895

Direct manipulation of 3-D objects through multimodal control

by

Z. Kazi, M. Salganicoff, M. Beitler, S. Chen, D. Chester and R. Foulds

1.0 Introduction

Direct manipulation Graphical User Interfaces (GUI) have become the predominant paradigm for human computer interfacing in the 1990's. The commercial successes of personal computer systems using direct manipulation such as Windows 3.1 tm, the MacOs tm and X Windows System have shown that such approaches to interaction are extremely effective and popular with users. Direct manipulation systems exploit intuitive common sense skills that are developed throughout the life of the individual by transferring them to operate on semantic task representations in a natural and transparent way. This allows individuals to utilize frequently used and highly developed skills such as those involved in manipulating real physical objects (e.g., the desktop metaphor) to learn to manipulate semantic task representations easily and with few syntactic constraints. The user can concentrate on task solutions, rather than expend cognitive effort in maintaining representations of the computer’s structures and processes and syntactic rules for forming commands. Direct manipulation GUIs have been shown to reduce learning time, error rates and to increase skill retention and subject satisfaction of users [Shneiderman, 1992].

Translating the direct manipulation metaphor into a 3-D domain reveals a new set of problems, especially for users with physical disabilities. While direct manipulation of graphical computer interfaces is now within the reach of many users with motor disabilities (e.g., the headmouse tm head movement controlled mouse, single switch control, voice recognition), a corresponding ability to carry out real direct manipulation on physical objects in their environment is lacking due to a variety of factors (e.g., spinal cord injuries, Multiple Sclerosis etc.). The physical limitations of motion range, coordination of movement and grasping, and lack of strength all contribute to a decreased ability to perform normal manual tasks. Fortunately, in principle, this loss may be compensated for by the use of assistive robots which may act on the user’s behalf in carrying out the direct manipulation.

The Multimodal User Supervised Interface and Intelligent Control (MUSIIC) project is working towards developing an assistive robot system that applies the direct manipulation metaphor by using a multimodal (speech and gesture) interface to allow people with disabilities to manipulate real world 3-D objects [Chen et al., 1994, Kazi et al., 1995b, Kazi et al., 1995a, Beitler et al., 1995b, Beitler et al., 1995a].

Our research involves a method and system which integrates human-computer interaction with reactive planning to operate a telerobot for use as an assistive device. The MUSIIC strategy is a novel approach for an intelligent assistive telerobotic system: speech-deictic gesture control integrated with a knowledge-driven reactive planner and a stereo-vision system. The system is intended to meet the needs of individuals with physical disabilities and operate in an unstructured environment, rather than in a structured workcell allowing the user considerable freedom and flexibility in terms of control and operating ease. The strategy utilizes a stereo-vision system to determine the three-dimensional shape and pose of objects and surfaces which are in the immediate environment, and provides an object-oriented knowledge base and planning system which superimposes information about common objects in the three-dimensional world. This approach allows the user to identify objects and tasks via a multimodal user interface which interprets their deictic gestures and speech inputs. The multimodal interface performs a critical disambiguation function by binding the spoken words to a locus in the physical work space. The spoken input is also used to supplant the need for general purpose object recognition. Instead, three-dimensional shape information is augmented by the user's spoken word which may also invoke the appropriate inheritance of object properties using the adopted hierarchical object-oriented representation scheme. To understand the intricacies and embodied meaning of the numerous modal inputs, we have also designed a graphical simulation of the multimodal environment. This simulation will allow us to study and better understand the interplay between the user and the MUSIIC system. Additionally, the simulated environment will be an integral part of the actual MUSIIC system by providing the user a visualization which depicts the planner's interpretation of the information gathered by the system. The MUSIIC system's ability to determine the superquadric shape representation of the scene from stereo vision enables the graphical simulation to dynamically model a variety of real world entities and objects.

A very simple illustration (Figure 1) describes how our proposed system functions in a real-world scenario.

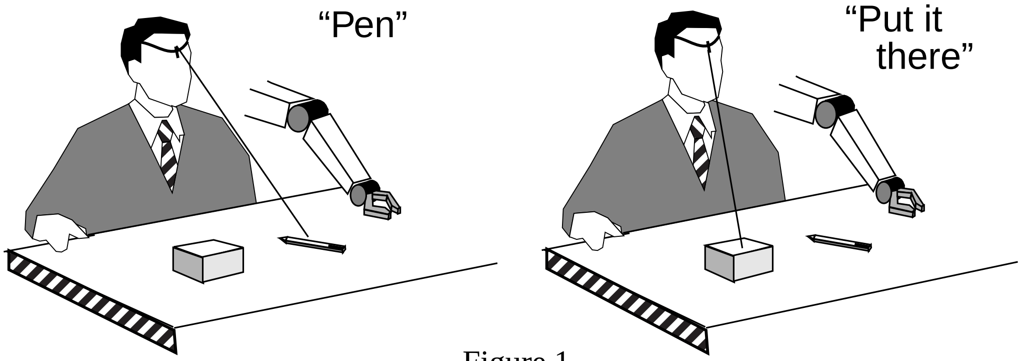

Figure 1

The user approaches a table on which there are a *pen* and a *box* The user points to the pen, and says, *pen*. The user points to the box and says *put it there*, indicating that the pen must be moved to the location *there*. The system then executes the user intentions.

1.1 Justification

A multimodal interface falls between two technical solutions - master/slave telemanipulation and autonomous robots. The former places significant physical demands on the user, the latter requires extreme structure in the world coupled with unrealized machine intelligence. A multimodal interface would allow the user to remain in the loop, while lessening the physical demands.

1.1.1 Towards a direct manipulation RUI (Robot User Interface) for three-dimension-

al unstructured environments

Traditional systems have used command line interfaces with voice-recognition to allow the user to instruct the robot on what actions are desired. This is a situation reminiscent of early user interfaces which allow dialogue through a syntax constrained command line interface. In the direct manipulation robot interface, the user concentrates on the task semantic representation and direct manipulation of physical objects, rather than the internal representation of the robot and computer controller. People without disabilities can easily identify, manipulate and transport objects while taking the environmental context into account because we have exceptional sensorimotor abilities. Employing a direct manipulation approach to the control of an assistive robot brings with it similar advantages to those available with direct manipulation two-dimensional GUI's where manipulation acts can be linked to processes internal to the computer or system being controlled. Certainly, a system that by-passes a fatigue inducing command line interface to an assistive robot and permits a natural human-task dialogue would be highly advantageous, considering that an assistive robot will have to be used almost constantly if it is to be cost-effective and worthwhile to the user.

Implementing a two-dimensional direct manipulation GUI can be a relatively complicated endeavor, but is significantly easier than a RUI which operates on physical objects located in the real unstructured three-dimensional world around the individual with disabilities. Object oriented development environments for two-dimensional GUIs abound, where screen controls (e.g. sliders and radio buttons) can be instantiated and carry with them associated interaction methods and responses to user input. In these development environments, the location and identity of all screen objects are known to the window manager and there is no ambiguity in sending events to the different screen objects. In a three-dimensional direct manipulation interface for unstructured environments, rather than having screen objects which are cursor addressable using a mouse, the user points to physical objects in the world via gesture, and specifies that certain manipulatory actions be performed on the objects. Analogous to screen objects, real objects have semantics and methods associated with them that define their shape, functions, size and allowable actions that can be performed on then, as well as requiring what the expected outcome of the actions will be. However, in order to operate the system must have the ability to quickly and reliably identify the objects and their position and orientation in the environment.

Enabling the computer and robot to be aware of the identity and pose of a physical object so that direct manipulation may be carried out by the robot system introduces significant perceptual and motor bottlenecks. The computer's internal representation of the domain must be updated with respect to the physical objects identity, shape, pose and location, as well as any constraints that might be associated with the manipulation of that particular object. These attributes are not immediately accessible as they are in the case of a window manager with screen objects and associated data structures through which they can be rapidly indexed. In principle, a highly reliable and rapid machine vision system would provide the necessary recognition and pose determination for objects, but this is currently far beyond the state of the art and even if it were, might involve significant costs in terms of processing time and system complexity.

Furthermore, actions specified by the user, such as the equivalent of dragging a screen object, would require grasping and transportation of real objects. Each of these processes is influenced by the world state, while in the case of a screen representation, such processes tend to be context independent. A human invokes highly sophisticated path planning abilities when planning trajectories for objects. Much work has been done in trajectory planning and plan synthesis in the robotics and AI planning communities, but practical, rapid and highly autonomous systems are still a long way from practical reality.

One approach towards mitigating the requirements for perceptual and planning systems to support a direct manipulation system is to utilize a multimodal interface to combine input evidence from a user dialogue. This permits perceptual and planning requirements of the system to be relaxed to the point where existing semi-autonomous techniques are sufficient to carry out tasks and make the system practical. By engaging in dialogue with the user in such a way that natural deictic gestures and voice input can be used to carry out a task, the system gains many of the advantages present in direct manipulation interfaces. The user can directly designate objects and locations in the environment around him/her, and use natural language to describe the desired actions on those objects and locations. By combining different modalities, rather than attempting to constrain dialogue to one modality, great simplification of processing can be accomplished, as has been demonstrated by several multimodal systems that have been developed for graphical user interfaces [Koons, 1994, Bolt, 1980]. This simplified processing allows for less delay in the processing of user interaction, which supports faster system response to user actions, that has been demonstrated to improve user task completion times and to result in less frustration [Shneiderman, 1992].

1.1.2 Supervisory control paradigm

A point of departure for our project is the supervisory control paradigm which is defined by Ferrel and Sheridan as a “control system where one or more human operators are intermittently programming and continually receiving information from a computer that itself closes an autonomous control loop through artificial end effectors and sensors to the controlled process or task environment” [Sheridan, 1992, Ferrel & Sheridan, 1967]. This allows the system to blend the flexible decision making and error-handling abilities of the human with the pre-programmed autonomous sub-goal satisfying ability of the machine, while still being monitored by a human operator.

A teleoperator is a machine (e.g., a robot) that projects an operator’s sensing and manipulation capabilities to a location remote from that person. Types of teleoperated systems may be broadly categorized into those using continuous control by the operator and those telerobots in which the human operator supervises the robot through a computer intermediary using an intermittent form of control. In other words, telerobotics is supervisory control of a teleoperator. Interfaces for intermittent supervisory control have taken on a variety of forms. However, many existing supervisory control systems have not used direct manipulations approaches; instead they rely on command-line type interfaces and requiring fairly good knowledge of the syntax and semantics of robot programming languages to set up tasks, which makes them difficult to use for people with little robotics experience.

1.1.3 Background work in supervisory and telemanipulatory control

Brooks developed an early and effective object-centered interface for supervisory control [Brooks, 1979]. His SUPERMAN (supervisory manual control) system provided for analogic continuous control of a telerobot as well as symbolic level identification of particular objects via keyboard input. The operator would carry out a specific set of operations on a given object. The manipulations would be stored relative to an object-centered coordinate frame in the form of a macro defined with respect to that object (which was given a symbolic name via keyboard input). Thus, the operator would train the system by identifying objects and providing manipulatory macros for subsequent execution relative to a new instantiation of a similar (or identical) object in a different location. The identification of objects and description of pose was done by the operator, which in turn invoked the appropriate manipulatory macro in that object's pose coordinate frame. The system was highly successful; empirical evaluations showed that operators generally experienced quicker completion times and fewer errors than complete continuous control without the benefits of supervisory control. Thus, this system demonstrated that the perceptual and reasoning abilities of a non-disabled operator could be effectively married with the semi-autonomous activities of the robot control system.

Schneider developed a “select and drag” interface for a telerobot in a planar task [Schneider & Cannon, 1989]. The interface allowed the operator to use a mouse to select an object to be transported by clicking on it, and then dragging the selected object to the desired location to provide for the selection of the destination location.

Cannon has developed “point and direct” interfaces for supervisory control of robots. Cannon [Cannon, 1992] used a three-dimensional point designation system for specifying positions and locations in a workspace by the alignment of two reticles from two camera views of the remote scene. The system supports a “put that there” level of syntax for binding objects and locations to actions. Subsequently Cannon has developed an augmented reality system to superimpose tool renderings on live video images of the remote scene [Cannon et al., 1994]. The virtual tools are guided through the use of a three-dimensional tracker and glove interface.

Funda *et al* have developed a teleprogramming paradigm for telemanipulation where *a priori* information about the task scene is used to generate a virtual environment with graphical and force feedback [Funda et al., 1992]. In the environment the user can directly manipulate three-dimensional graphical objects. The actions of the operator are parsed into symbolic actions on specific objects which are then transmitted to a remote site where registration between the virtual and real-environment takes place and the guarded actions are executed on the corresponding real objects. However, implementation of such a system requires significant overhead since scene models must be constructed each time the task scene changes and new objects appear.

Similarly, if the work place of the users is highly structured and static, then the system may be assured that given locations have given objects with given attributes, and indexing of object pose shape and size may be indirectly accessible through location or symbolic input [Leifer, 1992]. However, systems such as these suffer from low user acceptability due to their inflexibility [van der loos et al., 1990b, van der loos et al., 1990a]. Secondly, from a cost standpoint, significant effort and cost is incurred each time a new task must be programmed, such as in vocational environments, making such systems non cost-effective. What is desired is a highly flexible system that does not require programming *per se*, that leaves the locus of control with the user. The system should utilize its excellent perceptual, means-end reasoning and path planning abilities to flexibly and directly carry out the desires of the user. It is essential that the user can carry out this control without having to worry about the internal representations of the computer and the perceptual system present in the assistive robot.

A different approach to command-based robot operation was proposed by Harwin et al [Harwin et al., 1986]. A vision system viewed the robot’s workspace and was programmed to recognize bar codes that were printed on each object. By reading the barcodes and calculating the size and orientation of the barcode, the robot knew the location and orientation of every item. This was successful within a limited and structured environment. This system did not easily lend itself to a variety of locations and was not able to accommodate the needs of individuals with disabilities in unstructured environments. It did, however, demonstrate the dramatic reduction in *machine intelligence* that came by eliminating the need for the robot to perform object recognition and language understanding.

1.1.4 Human machine synergy

At the other extreme of robot control are the completely autonomous systems that perform with effectively no user supervision, the long elusive goal of the AI (Artificial Intelligence), robotics and machine vision communities. Unfortunately, this goal seems far from practical at this point, although many important incremental advances have been forthcoming in the past decades. Furthermore, absolute automation poses a set of problems stemming from incomplete *a priori* knowledge about the environment, hazards, and strategies of exploration, as well as from insufficient sensory information and the inherent inaccuracy in the robotic devices [Sheridan, 1992]

Therefore, continuing in the spirit of supervisory control, what one should strive for is a synergistic integration of the best abilities of both *“humans”* and *“machines”*. Humans excel in creativity, use of heuristics, flexibility and “common sense’, whereas machines excel in speed of computation, mechanical power and ability to persevere. While progress is being made in robotics in areas such as machine vision and sensor based control, there is much work that needs to be done in high level cognition and planning. We claim that the symbiosis of the high level cognitive abilities of the human, such as object recognition, high level planning, and event driven reactivity with the native skills of a robot can result in a human-robot system that will function better than both traditional robotic assistive systems and autonomous systems. We describe a system that can exploit the low-level machine perceptual and motor skills and excellent AI planning tools currently achievable, while allowing the user to concentrate on handling the problems that they are best suited for, namely high-level problem solving, object recognition, error handling and error recovery. By doing so, the cognitive load on the user is decreased, the system becomes more flexible, less fatiguing, and is ultimately a more effective assistant.

The rest of the paper is organized as follows. In section 2 we describe the architecture of the MUSIIC system. In section 3 we discuss the details of the multimodal interface. In section 4 we discuss pertinent issues in design followed by a summary in section 5.

2.0 MUSIIC

In this section we discuss both the implementation as well as the architecture of the MUSIIC system.

2.1 System Description

The previous sections lead naturally to a description of the essential components of the MUSIIC system. We require a ***planner*** that will interpret and satisfy user intentions. The planner is built upon ***object oriented knowledge bases*** that allow the users to manipulate objects that are either known or unknown to the system. A ***speech input*** system is needed for user inputs, and a ***gesture identification*** mechanism is necessary to obtain the user's deictic gesture inputs. An ***active stereo-vision*** system is necessary to provide a snap-shot of the domain; it returns object shapes, poses and location information without performing any object recognition. The vision system is also used to identify the focus of the user's deictic gesture, returning to the planner information about either an object or a location. The planner extracts user intentions from the combined speech and gesture input. It then develops a plan for execution on the world model built up from the *a priori* information contained in the knowledge bases, the real-time information obtained from the vision system, the sensory information obtained from the robot arm, as well as information previously extracted from the user dialog. Prior to execution, the system allows the user to preview and validate the planner's interpretation of user intentions via a 3-D graphically ***simulated environment***.

2.1.1 Hardware architecture

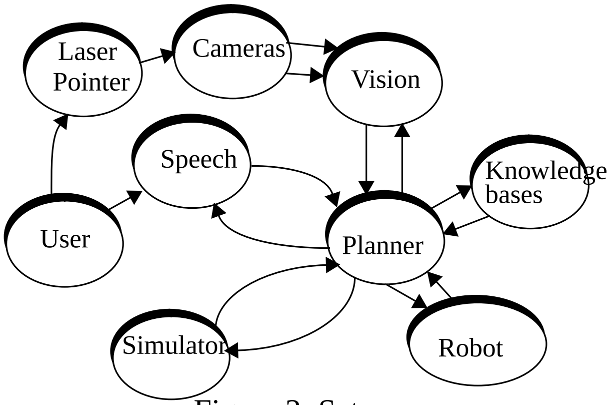

Figure 2: Set-up

The configuration of the interface system is depicted in Figure 2 and the system set-up is shown in Figure 3.

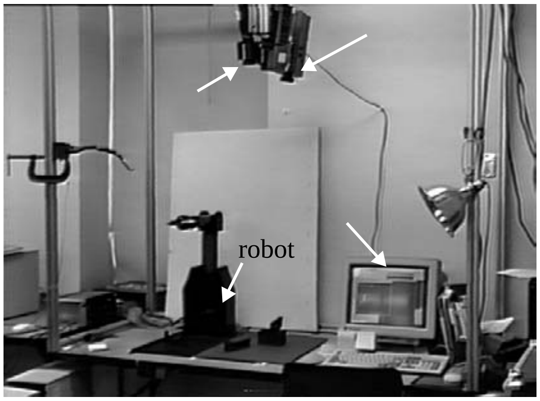

Figure 3: Physical Set-Up

The backbone computing machine for the vision interface is an SGI XS-24 IRIS Indigo computer. Pictures are taken by two CCD color cameras, model VCP-920 that have light-resolution 450 TV lines with 768X494 picture elements. Each camera is equipped with motorized TV zoom lens, model Computer M61212MSP. Cameras are connected to the SGI Galileo graphics board which provides up to three input channels. In this system we use two channels with s-video inputs. The Noesis Visilog-4 software package installed on the SGI machine is used as an image processing engine to assist developing the vision interface software. The speech system used is Dragon Dictate running on a PC. A six degree of freedom robot manipulator, Zebra ZERO, is employed as the manipulation tool. The planner and knowledge bases reside on a Sun Sparc 5. The simulated environment is run on anther SGI machine. Communications between the planner and the sub-systems are supported by the RPC (Remote Procedure Call) protocol.

2.1.2 The high level planner

We describe an architecture for task planning which incorporates a novel reactive planning mechanism where the user is an integral component of the planning mechanism. The planning mechanism is based on an object-oriented knowledge base incorporating in it the relaxed assumptions about the world that are essential for the mechanism to be practical in the real world and facilitating human-computer interaction as a means of providing reactive and replanning capabilities.

Reactivity is achieved in two ways. An autonomous runtime reactivity is obtained through sensor fusion. Sensory information from the vision system and force sensors will be used by the planner to obtain information for not only task planning but also to react to environment changes. Sensor uncertainty and computational complexity prevents having a totally sensor based reactive planning system, and hence user input is necessary for imparting the necessary reactivity.

Our hierarchical human-machine interface and object oriented representation allows the user to interact with the planning system at any level of the planning hierarchy, from low-level motion and grasp planning to high-level task planning of complex tasks such as feeding. The generic plans and specialized plans are supplemented by user interaction whenever incomplete information precludes the development of correct plans by taking over control of the planning mechanism or providing information to the knowledge bases to facilitate the development of a plan capable of handling a new or uncertain situation. Furthermore, incomplete sensory information may be supplemented by user input, enabling the planner to develop plans from its plan library without the need for extensive user intervention.

Given this underlying architecture, the system first determines what the user wants, and then makes plans to accomplish the task. As a consequence of insufficient information, uncertainty, advent of new information, or failure of a plan, the system engages in a dialogue with the user which enables the planner to revise its plans and actions.

2.1.2.1 The architecture of the planner

The basic architecture in brief is composed of three knowledge bases: A general knowledge base of objects (**WorldBase**), a knowledge base of objects in the actual domain of operation (**DomainBase**), and a knowledge base of plans (**PlanBase**). The planner uses the three knowledge bases and user/sensor provided feedback, to construct robot plans.

2.1.2.2 WorldBase

Objects are represented in an increasingly specialized sequence of object classes in an inheritance hierarchy. We have devised a four tiered hierarchy, where object classes become increasingly specialized from the top level hierarchy to the bottom level hierarchy (Figure 4). At the top level, we start with a generic abstract object which causes generic plans to be developed for objects about which we do not have exact information. The second level object classes are classed in terms of general shapes such as cylindrical, flat and spherical. This enables the planner to modify plans for approaching and grasping when more information is available about the object. The third level constitutes general representation of commonly used everyday objects, such as a "*cup*" or a "*can*", and at the bottom level we end up with actual objects in the domain whose attributes are fully specified.

Each object, depending on the degree of generalization, has a set of attributes that will assist the planner in developing correct plans. An initial investigation into the kind of tasks the robot might be called on to undertake prompts us to visualize a set of attributes which include shape, size, dimensions, weight, approach point, grasp points, constraints and plan fragments. The constraints and plan fragments attributes need to be described in a little more detail to explain the working of our model:

Constraints—Constraints may be placed on objects which further constrain low level robot operations such as approaching, grasping, and moving. For example, we may place a constraint on a cup such that the cup's orientation cannot be changed during transport to prevent spillage. However, constraints can be relaxed and these constraints are dependent on which action is being invoked upon the object. For example, in the case of the cup, the constraint about the fixed orientation is dependent upon whether the cup is empty of full

Plan Fragments—Another needed component are plan fragments that are incorporated into plans formed by the planner. Certain tasks may be specific to an object, and plan fragments for those tasks may be associated with the object in question in order to facilitate correct planning.

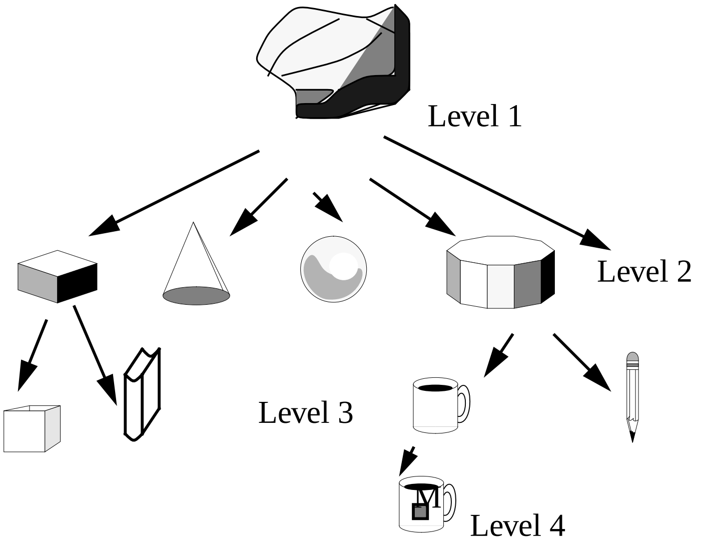

Figure 4 : Object Hierarchy Illustration

2.1.2.3 DomainBase

In addition to the knowledge base of objects, the system also maintains a knowledge base of objects that it sees in the domain, called the **DomainBase**. The objects in the domain contain additional attributes which are instantiated after objects have been identified by the system. Currently, amongst the attributes considered necessary are location and orientation, as well as attachment relationships to other objects and the workspace.

2.1.2.4 Object hierarchy user dialog illustration

A very simple example of the object hierarchy is shown below. Prior to interaction with the user, the system sets up the **DomainBase** as a collection of *blobs* of different sizes and shapes, with only the position with respect to the world origin being known. The *blob* world image is obtained from the vision system, and size and location parameters are instantiated in the **DomainBase** from the information obtained by the vision system. *We do not do any object recognition.* Based on the premise that the user is in the planning loop and can combine inputs from multiple modalities, the user points to a *blob* and identifies it to the system. For example, she may point to a specific blob using a co-verbal gesture and inform the system that this is a *cup*. The system then updates the attribute slots of the *blob* with attributes that it obtains from the **WorldBase**. The user may also identify the *blob* as a specific object, such as *my-cup*. In such a case, the system is aware of a specific object in the **WorldBase** which is known as *my-cup*. The *blob* in the **DomainBase** is replaced by the exact *my-cup* that the system knows about, and the attributes of *my-cup* in the **DomainBase** are instantiated from the **WorldBase** and information obtained from the vision system. It is entirely possible that the user may not have identified any specific *blob*, and the system then is only aware of the general shape, and identify the blob at a certain degree of generalization.

2.1.2.5 PlanBase

The plan knowledge base, **PlanBase**, is a collection of **STRIPS**-like plans [Sacerdoti, 1975, Sacerdoti, 1977], and the planner is based on a modified STRIPS-like planning mechanism. The main difference between conventional **STRIPS**-like planning and our proposed system is that we take full advantage of the underlying object oriented representation of the domain objects, which drives the planning mechanism. Plans in this model are considered as general templates of actions, where plan parameters are instantiated from both the WorldBase and the DomainBase during the planning process. For example, the constraints slot for a *Move* action might contain the slot *Object-constraints*. This implies that this slot parameter is going to be filled up from the constraints field of the object on which the action is being invoked. In the case of the cup example previously illustrated, the constraint that the cup must be maintained in a certain orientation is used to instantiate the *constraint* slot of the *Move* action. The constraints instantiated from the object in question are added to the set of constraints already present. Sometimes, some of the constraints obtained from the objects themselves may be in direct contradiction to constraints already present in the action being invoked. When that happens, the constraints obtained from the object override default constraints in the action body.

2.1.2.6 Handling exceptions

Another way in which the object oriented paradigm has extended the classical **STRIPS** planning mechanism is described here. As mentioned previously, the body of an action may contain further subactions into which the actions may be decomposed. This facilitates hierarchical planning, one of the essential features of a planning system.

However, certain tasks that can be handled generally for most objects may not be applicable to certain objects in the real world. Suppose we have an appliance that is used often in the domain of the user. The instrument has a peculiar shape and must be picked up from a specific point. To approach the grasp-point, it may not be possible to just simply specify a certain approach point and assume that the robotic arm will then be able to pick up that appliance. The approach path may be convoluted and hence there must be some way to specify such an atypical case in our planning system. This is done by the use of the plan-fragment associated with an object. In a manner similar to the way *action* slots are filled, depending on the object on which the actions are invoked, *subaction* slots are also filled, if so specified, from the object's *plan-fragments* slot.

Thus we see that this integration of knowledge base planning with an object oriented approach allows us to use general plans whenever we can. Additionally, this method will allow us to develop plans for specific objects which are peculiar to the domain without the need to perform computationally expensive operations. The object abstraction hierarchy allows us to abstract out the general features of an action and invoke them on objects about which the knowledge bases might not have any information. It also allows us to view an action as a single template that is applicable to many kinds of objects instead of as a set of actions, each applicable to only one specific object.

2.1.3 Vision

For our multimodal system, the vision requirement is to provide the knowledge based planning system with parameterized shape and pose information of the objects in the immediate environment. This information can then be used to fill slots in the object oriented representation and support both the system planning and simulation activities. The vision processing proceeds in three phases: extraction of highly precise 3-D point information using a calibrated line-based stereo matching algorithm, segmentation of the point sets into object-based sets, and non-linear minimization to fit parameterized shapes to respective objects in the scene. A feature-based matching algorithm is used for this application. To reduce the false extraction rate a high intensity structured-light source with parallel stripes is employed in this design. The distorted light patterns in the images can be easily extracted and processed. To recover the 3D contour of the objects the vision system needs to find the correspondence of the distorted patterns in two images. We have adopted the straight line pattern since it naturally incorporates the figural continuity constraint [Chai & Tsai, 1993]. A line-segment pair-match scheme is developed based on the geometric characteristics of the features obtained from the images.

Images are taken by two CCD cameras. The light source is generated by a slide projector in a form of light-stripes or grid. Existing stereo vision techniques for depth extraction are classified into several categories:

- full scale nonlinear optimization method,
- two plane method
- linear least-squares method [Hall & Tio, 1982]

The linear least-squares method is adopted in this project [Kazi et al., 1995a, Kazi et al., 1995b].

2.1.3.1 Calibration

Before the vision system can be used to extract points from the stereo image pairs, a precise calibration must be achieved to ensure that the disparity measurements resulting from the edge matching process can be triangulated to yield the true three-dimensional depth.

2.1.3.2 Line-segment pair matching process

The objective of stereo vision is to recover 3D information about the objects in the work environment using images taken from different viewpoints. The most essential and most difficult procedure in stereo vision is feature-matching. The purpose of the match process is to find the correspondence among the features extracted from two images. The difficulty of image match problem stems from factors such as image variations due to different perspective projections, occluded features and the source of lighting. Researchers in this field have been developing various algorithms in the past two decades. Basically, these algorithms can be classified into two major categories: area-based (intensity level as the feature) and feature-based (semantic features with specific spatial geometry) techniques [Barnard & Fischer, 1984]. Early representative works [Barnard & Thompson, 1980, Medioni & Nevatia, 1984, Price, 1986] used the relaxation labeling technique to solve the stereo image matching problem. More recent developments incorporating structural information between image entities in addition to entity properties solve the correspondence problem [Boyer & Kak, 1989, Horaud & Skordas, 1989, Matsuyama et al., 1984]. It should be noted that there is no currently unified approach to the stereo correspondence problem; it is very much application dependent. Details of both the calibration and line-segment pair matching schema are given in [Kazi et al., 1995b].

2.1.3.3 Segmentation and shape fitting

The purpose of the shape extraction system is to derive a set of shapes from a large number of point-wise measurements on the surfaces of the different objects in the scene that were derived by the stereo matching algorithm. Numerous representations are currently used for shape representations in both the CAD and vision communities, such as spline surfaces, generalized cones and superquadrics. Superquadrics are a superset of the class of ellipsoids which can represent and approximate many shapes from spheres to cubes and cylinders that occur in man-made environments (in fact, superquadrics were originated by the Danish designer Piet Hein.) [Barr, 1981]. Superquadrics provide two major advantages: a well developed mathematical foundation for their recovery from sets of range points [Bajcsy & Solina, 1987], and a concise shape description appropriate for planning, graphical display, and manipulation activities that occur in a planner and a graphically simulated world.

The shape extraction process consists of thresholding, segmentation and shape fitting of each respective point group. Since the height of the surface of support of the objects can be known *a-priori*, a threshold height may be set for the purpose of foreground-background segmentation. Once the thresholding is complete, a point-set clustering is performed on the single set of points that have been labelled as foreground points since there may be multiple objects in the scene. A nearest-neighbor metric is used to bottom-up cluster the point-set into subsets of connected-components according to a scaled Euclidean distance metric. The scaling allows for selectable merging distance thresholds in each of the orthogonal directions. Each resulting connected component point-subset then corresponds to an object in the scene.

Once the individual point sets have been clustered, the shape fitting process may be run on each individual point-set. The shape fitting process computes the shape parameters which control the shape, size and location of each superquadric shape. We use a non-linear minimization technique [Bajcsy & Solina, 1987] to rapidly determine the set of shape parameters that best fit the raw 3-D points measured. The resulting parameters then describe the positions, orientations and shapes of each of the different objects in the environment so that the planning and simulation systems may exploit the resulting shape and position representations [Beitler et al., 1995b]. Figure 5 shows the raw data, which is then segmented and fitted resulting in the approximation superquadric illustrated in Figure 6.

2.1.4 Simulated environment

We are developing a simulation environment

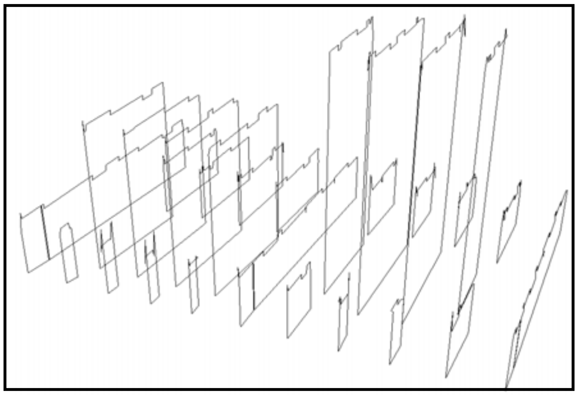

Figure 5: Recovered 3-D Points from Structured Light Stereo Line Matching

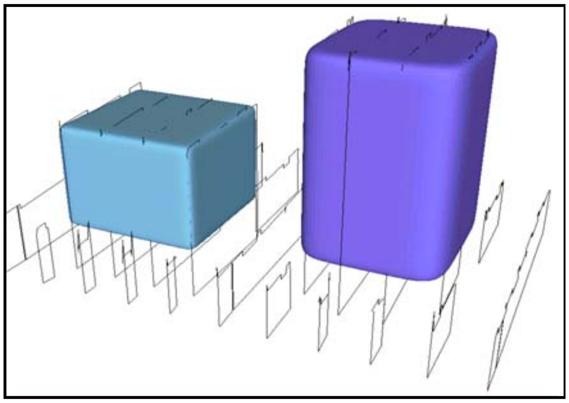

Figure 6: Point Segmentation and Resulting Object Shape Fitting

that will allow us to investigate, in a low risk fashion, the use of the multiple modalities of the user to control a rehabilitation robot. The type of simulation we are using has been referred to as a "fish-tank" environment, in which the individual feels that he is on the outside looking in through the side of a fish-tank (monitor screen) [Ware & Jessome, 1988]. This simulation models not only the robot and the domain but also the interplay between user intentions and the robot's perception of these intentions. This simulation mechanism has been developed using JACK [Badler et al., 1993].

A multimodal control system that can extract the embodied meaning of the numerous modal inputs and can properly respond to directives from a user depends heavily on having an understanding of the user's perception of the depth, distance, orientation and configuration of objects in the operating domain. It is important to note that our system does not require the user to provide information about the depth, distance, orientation and configuration of objects, but a mutual understanding of the user's perception of these features, and of the planning system's intentions is necessary to insure that tasks are carried out as the user intended. An important objective of the simulated multimodal environment is to allow our research team to rapidly implement and experiment with different methods of interpreting the discourse and gesture information. The simulated environment will also allow us to experiment with different techniques for combining the results of gesture and speech to extract their joint meaning [Beitler et al., 1995a]. The facets of multimodal control that we hope to better understand through the simulated multimodal environment are:

- User perception of object location and orientation
- User methods of interacting with objects and the robot
- Proper interpretation of the user's speech and gestural inputs
- Feedback about the multimodal control system's interpretation of the user's intentions
- Determination of a level of automation that allows the user the best control and flexibility
- User's insights into the system's possibly incomplete or erroneous scene representation
- Plan preview and error replay

An additional issue which the simulated environment will help to address is user safety. When the user commands the multimodal control system, the system is expected to complete that task without injuring the user or damaging the objects it is manipulating. To provide feedback to the user about the plans of the system, the simulated environment will be incorporated into the multimodal control system during its actual operation. The simulated environment will inform the user of the system's plans and interpretations of the world by showing a preview of what the system intends to do. Using the simulated environment to show a preview is very important because when the user entrusts the multimodal control system with a task, the user is "trusting" that the task will be performed correctly. Every time the user issues a command the user is also taking a "risk" that the system can do the job correctly [Foner, 1993]. Providing the user with a visual preview of the intention of the multimodal control system will effectively strengthen the "trust" between the user and the multimodal control system.

Both the MUSIIC system and the user need the ability to be able to communicate and understand each other. The simulated environment will help increase the mutual intelligibility of the interaction process by allowing the system to make it's own intention and knowledge apparent to the user, while also giving the user means of critiquing the plans developed autonomously by the planner.

3.0 The multimodal interface

Researchers have proposed a number of systems which investigate alternate modes of human-computer interaction in addition to speech and vision based ones. Work has been carried out in using gestures and hand pointing as a mode of man-machine interface. In some systems, researchers have required the users to use hand gloves [Cipolla et al., 1992, Fukimoto et al., 1992], while others require calibration for each individuals hand shapes and gestures [Wiemer & Ganapathy, 1989]. Cipolla et al. report preliminary work on a gesture-based interface for robot control [Cipolla et al., 1992]. Their system requires no physical contact with the operator, but uses un-calibrated stereo vision with active contours to track the position and pointing direction of a hand. Pook describes a deictic gesture based tele-assistance system for direct control of a telerobot, although the system lacks a perceptual component [Pook, 1994]. Funda et al. describe a teleprogramming approach which extracts user intentions from interaction with a virtual model of a remote environment, but their system requires an *a priori* 3-D model of the remote scene [Funda et al., 1992].

Work is also being done in attempting to extend this concept by using multiple modes of human-machine interfacing. We previously discussed the work of Bolt, Cannon and Paul in extending the metaphor of control of a robot into multiple modes. MUSIIC extends the combined deictic gesture and spoken word of Bolt to true 3-D environments manipulated by a robot. The combination of spoken language along with pointing performs a critical disambiguation function. It binds the spoken words in terms of nouns and actions to a locus in the physical workspace. The gesture control and the spoken input are used to make a general purpose object recognition module unnecessary. Instead, 3-D shape information is augmented by the user's spoken word which may also invoke the appropriate inheritance of object properties using the adopted hierarchical object-oriented representation scheme.

In the introduction we argued how using a multimodal interface to combine input evidence from a user dialogue mitigates the requirements for perceptual and planning systems to support direct manipulation. In the following sections we discuss the multimodal control input language.

3.1 Semantic interpretation for robot con-

trol

In order to devise a practical command input interpretation mechanism we restricted both the nature of our speech input as well as our gesture input.

3.1.1 Speech

Consider the user command:

`Put the book on the table`

The user is not required to spell out the actual procedures needed to satisfy her intentions, however these expressed intentions carry along with them conditions that may restrict the procedures that are invoked. While these conditions are not given in advance, they depend on the context in which the procedures are being invoked. Satisfaction of the user's intention entails the satisfaction of the equally important associated conditions that were not necessarily specified directly by the user. Therefore, it becomes very important to be able to unambiguously extract user intentions.

While a fully fledged natural language system combined with a state-of-the-art gesture recognition mechanism may allow the user more expressive power, the state-of-the-art in these two areas makes this a distant goal. At the same time, the requirements of the domain places some constraints on the choice of modalities and the degree of freedom in expressing user intentions. A multimodal combination of speech and pointing is a better alternative for use as an assistive device, where the input speech is a restrictive sub-set of natural language, a pseudo-natural language (PNL). We then can apply model-based procedural semantics [Crangle et al., 1988], where words are interpreted as procedures that operate on the model of the robot's physical environment. One of the major questions in procedural semantics has been the choice of candidate procedures. Without any constraints, no procedural account will be preferred over another and there will not be any shortage of candidate procedures. The restrictive PNL and the finite set of manipulatable objects in the robots domain provide this much needed set of constraints.

3.1.2 Gesture

Similarly, the needs of users with disabilities also restrict the choice of gestures. Our gesture of choice is deictic gesture, which is simply pointing. In the general case, not only does pointing have the obvious function of indicating objects and events in the real world, but it also plays a role in focusing on events/objects/actions that may not be objectively present [McNeill, 1982]. The choice of deictic gestures allows us to use any number of devices, not restricted to the hand, to identify the user's focus. While our research is investigating the use of a laser pointer to identify the user's focus of intentions, any device that is able to indicate a domain object can be used, such as eye tracking systems, mouse on a control panel, etc.

3.1.3 Combining speech and gesture

Like natural languages, gestures convey meanings. While their expressiveness is not inferior to natural languages, the methods used by gestures are fundamentally different from that of language. Segmentation and linearization to form a hierarchically structured string of words that are the essential feature of a linguistic system is based on the fact that language can vary only along the temporal dimension. Gestures are different in every way. McNeill describes a number of ways in which gestures are different [McNeill, 1982].

- Gestures are global-synthetic
- Gestures are combinatoric
- Gestures have no standards of form

- Gestures have no duality of patterns

These inherent differences makes gesture identification a very difficult task. However, while gestures and speech differ from each other in a number of fundamental ways, they are also closely linked in many ways.

- Gestures occur during speech
- Gestures and speech are semantically and pragmatically co-expressive
- Gestures and speech are synchronous

Restricting our choice of gestures to pointing gestures only, allows us to use the above properties to extract user intentions in an unambiguous way. We are using pointing gestures to identify the user's focus of attention, to indicate either an object or a location. Currently, speech deictics "that" and "there" are being used in conjunction with pointing to identify the user's focus. The interpretation process must be able to capture the user's actions in speech and gesture within the domain of operation and then attempt to match them to elements in the system's domain knowledge base. We are able to extract the combined user intention by the use of time-stamps that allow us to identify which object or which location was the focus of intention during the user's deictic utterances. Each word is tagged with a time stamp, and the vision system is continuously scanning the world and storing a history of points identified by the gesture (in our case the laser pointer). Depending upon whether the speech deictic was a "that" or a "there", the procedures encoded with each word then returns either an object or location respectively. The required action is then invoked upon the returned values.

3.2 Semantics of the multimodal interface

Let us investigate a typical MUSIIC instruction for the robot: (the words in square brackets imply speech combined with a pointing gesture)

```
    Put [that] [there].
```

Analyzing the components:

Put -> ***TASK*** specification; Semantic analog to a Verb in Natural Language

[that]-> Deictic that gets instantiated to an ***THING***. Analogous to a Subject in Natural Language.

[there]-> Deictic that gets instantiated to a ***LOCATION***.

Mapping the major syntactic components to their corresponding semantic elements in the current implementation we obtain:

Put:->***TASK***
that->***THING (TASK-FOCUS)***
there:->***LOCATION (DESTINATION)***

From a pure speech input, we may have an instruction such as:

```
    Push slowly the blue book
next to the red cup 2 feet
towards me.
```

Mapping the major syntactic components of this sentence to their corresponding semantic elements, we obtain:

Push:->***TASK***
slowly:-> ***TASK-QUALIFIER***
the blue book->***THING (TASK-FOCUS)***
next to the red cup:->***LOCATION (SOURCE)***
2 feet:->***QUANTITY***
towards me:->***LOCATION (DESTINATION)***

In essence a typical instruction would have the following semantic format:

TASK
TASK-QUALIFIER
TASK-FOCUS
SOURCE-LOCATION
QUANTITY
DEST-LOCATION

While a complete natural language mechanism is not desired at this point, a syntactic structure that simulates to a certain extent the syntax of natural language (though restricted) would make the user feel more comfortable with the system.

The Semantic Units (SU) being used are:

TASK: The action that is to be performed.

TASK-QUALIFIER: Qualifying how the action is going to be invoked. Slowly and fast.

TASK-FOCUS: ***TASK*** being invoked on this ***THING***

SOURCE-LOCATION: Of type ***LOCATION***

QUANTITY: Spatial/Temporal duration of the ***TASK***

DESTINATION-LOCATION: Of type ***LOCATION***

THING: Is an ***SU*** similar to a noun-phrase in Natural Language. Elements of ***THING*** are, {***ART***}[1], {***ADJ***}[1] and {***OBJECT***}

ART: a, an, the, that, this
ADJ: Object quantifier. Properties such as weight, color, size, surface.
OBJECT: The actual manipulatable object. Both abstract as well as specific.

LOCATION: An ***SU*** that maps to an ***OBJECT*** position in the world with respect to a certain frame of reference. What is also needed is a location function (***LF***) to define locational relationships such as “in”, “inside”, “above”, “below” etc. The ***LF*** takes the locational relationship and a ***THING*** and maps it to a ***LOCATION***.

QUANTITY: A spatial or temporal quantity.

4.0 Discussion

The on-going research on the MUSIIC project, developing an intuitive multimodal RUI for an assistive robot, leads to some very interesting problems that need to be addressed. These problems stem from many different factors; from general principles, to problems that stem from interaction with a three dimensional unstructured world, to problems that stem from the fact that the focus is on the use of this RUI for the control of an assistive robot by persons with disabilities, which puts further restrictions on the design.

As with any human-computer system, the design of the robot user interface [Leifer, 1992] for an assistive robot is driven by many of the same considerations as those that drive the design of graphical two-dimensional user interfaces. However, interfaces for the control of robots have many requirements that differ from those interfaces for non-embedded computers. Many of these stem from the that fact the robot is an electromechanical system capable of generating large forces and represents danger, both to objects and individuals in the task site. This is especially true if the operator and the robot are in close quarters, such as when the robot is used for assistive feeding.

Since the robot operates in the real world, it must satisfy many existing spatial and temporal constraints in order to successfully carry out manipulation tasks, prevent errors and ensure the safety of the operator and other

[1] Optional

individuals in the area around her. The notion of error prevention is especially important, since for many actions carried out by robots in the world, there is no recourse to an “undo” action [Leifer, 1992].

Another important principle of user interface design that is supported by multimodal direct manipulation interfaces for assistive and telerobotics is that of ease of action reversal. This is usually available with direct manipulations GUI's through the universal undo feature, whereby a user may perform the inverse of the most recent action. Obviously, with robot user interfaces, undo may only be feasible in certain situations. One way around this is to allow for preview of the systems actions before they are actually done, therefore, the user can view the outcome of an action in a non-destructive fashion, and prevent it from actually being executed by the robot if its outcome is undesirable. We support such a capability through a plan preview mechanism which allows a user to view a 3-D graphical simulation of the outcome of her gestural and verbal dialogue with the system.

We address this very important problem at different levels. The simulation mechanism described previously is one way in which we are enabling an “undo” option. The correct interpretation of user intentions is previewed before the actual execution. This allows the user to fine tune instructions if what the system intends to do isn’t what the user desired.

The planning mechanism is also endowed with a reactive component, with reactivity being achieved in two ways. An autonomous runtime reactivity is obtained through sensor fusion. Sensory information from the vision system, force sensors, etc. will be used by the planner to obtain information for not only task planning but also to react to environment changes. However, there are two fundamental limitations to having a totally sensor based reactive planning system:

- Uncertainty: Sensors can only return limited or incomplete information about the environment. Decisions as to what action to perform on what object are then made under conditions of uncertainty leading to possible failures.
- Computational Limitations: Sensor interaction is bounded by computational limitations. The most effective actions are those which are sufficient for the problem at hand and require as little effort as possible.

While these factors prevent us from having a totally sensor based reactive system, we overcome these restrictions by a hierarchical human-machine interface. The ability to interact with the planning system at any level of the planning hierarchy, from low level motion and grasp planning to high-level task planning of complex tasks allows the user to take over the planning process and supplement the autonomously developed plans.

4.1 Future directions

Our system is still very much a work in progress. In the following subsections, we describe a set of scenarios that illustrate how MUSIIC would perform in increasingly difficult situations as a means of illustrating the problems and issues that we have addressed.

4.2 Preliminaries

A partial description of the contents of the knowledge base is provided to facilitate the description. The system has in its knowledge base knowledge about top level objects such as cylinders, cubes, flat objects etc. In its knowledge base exists information about a specific cup *my-cup* that the user often uses. The *my-cup* object is hence fully instantiated. The exact dimensions, approach points, grasp points amongst others are fully specified. This *my-cup* object is derived from a more generic *cup* object. At this level of generalization, dimensions are given in terms of ranges and possible values. Furthermore, a constraint is placed on the *cup* object, which states that the *cup* must always be kept at a certain orientation. However, this constraint must be over-ridden if the user wants to pour something from the *cup* and such constraints can be over-ridden by the action being invoked on the object. Furthermore, in the plan library are top level actions such as *pour*, as well as more primitive actions such as *pick*, *put*, and *move*. Associated with each object are plan fragments that affect the actions which can be performed on them. These are placed in the plan fragment slots of the object attributes. This is so because not all plans can be made generic enough. For example, in the case of *my-cup*, the pick operation for a cup is very specific. The cup must be grasped by the handle and such an action is specific to only *cup* objects. So when the *pick* action is invoked on a *cup* object, the plan derives a sub-task from the object itself which determines how exactly the object might be picked. Given the object oriented nature of both objects and actions in our domain, this is easily facilitated.

4.3 Known objects

In this scenario, the user approaches his/her own desk where objects are routinely used and are familiar. These objects may be present in the knowledge base. The vision system surveys the scene, and computes a three dimensional surface (the orientation of this surface will be dependent upon the position of the user's wheelchair). The user points to the user's cup and identifies it as a known object with the word *my-cup*. This tells the system about its weight, its dimensions, and the approach path to be taken by the robot. The user says *move*. The user then points to a surface on the table and says *there*. From the information that the planner derives from *my-cup*'s parent class, the planner will then calculate the path that needs to be followed to place *my-cup* there, while maintaining the constraint that the *cup* must be kept at a certain orientation.

4.4 Unfamiliar environment

The user in a wheelchair equipped with a portable robot and its vision system approaches a desk. There are objects on the desk with which the system is not completely familiar. After the vision processing, the user points to a *cup* on the desk and identifies it as a *cup*. The system instantiates its world base from knowledge about the *cup* object. Now if the user then gives the same instruction as in the previous illustration, the planner would be able to plan the correct path on which the *cup* must be moved.

4.5 Plan adaptation or unknown object

In this modified scenario the user again approaches an unfamiliar environment. After the vision processing, the user points to a mug and tells the system that this is a *mug* object. The system was previously unaware of a *mug* object in its knowledge base. If the user wants to now pick and move the *mug* she can do one of four things.

She can load up the knowledge base with information regarding the *mug* object so that the system is able to handle operations on the *mug*.

Secondly, she can inform the system that a *mug* is a *cup-like* object and that it derives from a *cylinder* type object. When the user invokes a pick command on the *mug*, the system then generalizes the *pick* command applicable to a *cup* object and uses it on the mug. From the knowledge base the system is able to infer that a *cup* must be picked up from the handle, and the system then attempts to determine the location of the handle for the *mug* in order to ascertain what the approach points and the grasp points are going to be. Based on the vision system information and the generalization of the *pick* operation, the system then instantiates the parameters for the *mug* object so that next time it has no need to generalize. Further attributes for the object, such as weight etc., can be added during the actual process of executing the pick operation.

Thirdly, the user simply informs the system that the object is a *mug* and instructs it to pick the mug up. This time the system uses information gathered from the vision system to determine a suitable approach point and grasp point (this may not necessarily be the mug handle) and initiates the action with the gripper open wide enough to grasp, and uses the force sensors in its fingers to grasp the *mug*. This is an example of the most abstracted example of a *pick* operation in our plan knowledge base.

Fourthly, the user may direct the movements of the arm, in a cartesian control method via the 3-D graphically simulated environment. This would be accomplished by moving a 3-D crosshair to the desired location and orientation of the gripper and then commanding the system to move to the crosshair location.

Fifthly, the user may directly use a joint control method to manipulate the arm into the proper orientation to grasp the object. During all of these operation scenarios the system instantiates the attributes to the objects for later reuse.

4.6 Reactivity and user interaction

In a similar scenario as described in the preceding sub-sections the user instructs the arm to pick a *cup* and place it on the table at a certain location. At the end of the execution phase, the system ascertains that the *cup* which is still in its grasp is not quite touching the surface of the table. This determination is made through sensory information such as force feedback. When this occurs, the planner then generates a plan which will allow the cup to be placed on the table, while continually receiving force feedback to determine the successful completion of the task. In a more catastrophic scenario, the *cup* falls out of the grasp of the manipulator. The fact that the *cup* has fallen can be detected by the arm, but further execution of the plan is impossible and the system then needs to interact with the human user. The user may then determine whether the goal needs to be satisfied given the catastrophic event. In the event that the goal still needs to be satisfied, the system has to rescan the work space and the user needs to identify the new location of the dropped *cup* so that the planner can generate a new plan.

5.0 Conclusion

Human intervention as well as an intelligent planning mechanism are essential features of a practical telerobotic system. We believe our multimodal RUI is not only an intuitive interface for interaction with a three-dimensional unstructured world, but it also allows the man-machine synergy that is necessary for practical manipulation in a real world environment. Our novel approach of gesture-speech based human-machine interfacing enables our system to make realistic plans in a domain where we have to deal with uncertainty and incomplete information.

6.0 Acknowledgment

Work on this project is supported by the Rehabilitation Engineering Research Center on Rehabilitation Robotics, National Institute on Disabilities and Rehabilitation Research Grant #H133E30013, Rehabilitation Services Administration Grant #H129E20006 and Nemours Research Programs.

7.0 References

Badler, N. I., Phillips, C. B., & Webber, B. L. (1993). *Simulating humans*. Oxford University Press.

Bajcsy, R. & Solina, F. (1987). Three dimensional object representation revisited. In *First International Conference on Computer Vision* (pp. 231–240).: IEEE.

Barnard, D. T. & Fischer, M. A. (1984). Computational stereo. *ACM Computing Survey*, *14*(4), 553–572.

Barnard, D. T. & Thompson, W. B. (1980). Disparity analysis of images. *IEEE Transactions on Pattern Anal. Mach. Intell.*, *2*(4), 333–340.

Barr, A. (1981). Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, *1*(1), 11–23.

Beitler, M., Foulds, R., Kazi, Z., Chester, D., Chen, S., & Salganicoff, M. (1995a). A simulated environment of a multimodal user interface for a robot. In *RESNA 1995 Annual Conference* (pp. 490–492). Vancouver, Canada: RESNA press.

Beitler, M., Kazi, Z., Salganicoff, M., Foulds, R., Chen, S., & Chester, D. (1995b). Multimodal user supervised interface and intelligent control (MUSIIC). In *AAAI 1995 Fall Symposium Series on Embodied Language and Action* MIT, Cambridge, Massachussetts.

Bolt, R. A. (1980). Put that there: Voice and gesture at the graphics interface. *Computer Graphics*, *14*(3), 262–270.

Boyer, K. L. & Kak, A. C. (1989). Structural stereopsis for 3d vision. *IEEE Trans. on Pattern anal. Mach. Intell.*, *11*(11), 1168–1180.

Brooks, T. (1979). SUPERMAN: a system for supervisory manipulation and the study of human-computer computer interaction. Unpublished master's thesis, MIT, Cambridge, MA.

Cannon, D. (1992). *Point and direct telerobotics: Object level strategic supervision in unstructured human-machine interface*. Unpublished doctoral dissertation, Stanford University, Department of Mechanical Engineering.

Cannon, D., Thomas, G., Wang, C., & Kesavadas, T. (1994). Virtual reality based point-and-direct robotic system with instrumented glove. *Internation Journal of Industrial Engineering – Applications and Practice*, *1*(2), 139–148.

Chai, S. & Tsai, W. (1993). Line segment matching for 3d /computer vision using a new iteration scheme. *Machine Vision and Applications*, *6*, 191–205.

Chen, S., Kazi, Z., Foulds, R., & Chester, D. (1994). Multi-modal direction of a robot by individuals with a significant disabilities. In *Second International Conference on Rehabilitation Robotics 1994* (pp. 55–64).

Cipolla, R., Okamotot, Y., & Kuno, Y. (1992). Qualitative visual interpretation of hand gestures using motion parallax. In *IAPR Workshop on Machine Vision Applications* (pp. 477–482).

Crangle, C., Liang, L., Suppes, P., & Barlow, M. (1988). Using english to instruct a robotic aid: An experiment in an office-like environment. In *Proc. of the International Conf. of the Association for the Advancement of Rehabilitation Technology* (pp. 466–467).

Ferrel, W. R. & Sheridan, T. B. (1967). Supervisory control of remote manipulation. *IEEE Spectrum*, *4*(10), 81–88.

Foner, L. N. (1993). *What's an agent anyway? a sociological case study*. E-15 305, MIT Media Laboratory, 20 Ames Street, Cambridge, MA 02139.

Fukimoto, M., Mase, K., & Suenga, Y. (1992). Realtime detection of pointing action of a glove free interface. *Proc. IAPR Workshop on Machine Vision Applications*, (pp. 473–476).

Funda, J., Lindsay, T. S., & Paul, R. P. (1992). Teleprogramming: Towards time-invariant telemanipulation. *Presence*, *1*(1), 29–44.

Hall, E. L. & Tio, J. (1982). Measuring curved surfaces for robot vision. *Computer*, (pp. 42–45).

Harwin, W., Ginige, A., & Jackson, R. (1986). A potential application in early education and a possible role for a vision system in a workstation based robotic aid for physically disabled persons. In R. Foulds (Ed.), *Interactive robotic aids-one option for independent living: An international perspective*, volume Monograph 37 (pp. 18–23). World Rehabilitation Fund.

Horaud, R. & Skordas, T. (1989). Stereo correspondence through feature grouping and maximal cliques. *IEEE Trans. on Pattern anal. Mach. Intell.*, *11*(11), 1168–1180.

Kazi, Z., Beitler, M., Salganicoff, M., Chen, S., Chester, D., & Foulds, R. (1995a). Intelligent telerobotic assistant for people with disabilities. In *SPIE's*

International Symposium on Intelligent Systems: Telemanipulator and Telepresence Technologies II: SPIE.

Kazi, Z., Beitler, M., Salganicoff, M., Chen, S., Chester, D., & Foulds, R. (1995b). Multimodal user supervised interface and intelligent control (MUSIIC) for assistive robots. In *1995 IJCAI workshop on Developing AI Applications for the Disabled* (pp. 47–58).

Koons, D. P. (1994). Capturing and interpreting multimodal descriptions with multiple representations. In *Intelligent Multi-Media Multi-Modal Systems*.

Leifer, L. (1992). Rui: factoring the robot user interface. In *Proceedings of the Rehabilitation Engineering Society of North America (RESNA) conference*.

Matsuyama, T., Arita, H., & Nagao, M. (1984). Structural matching of line drawing using geometrical relationship between line segments. *Computer Graphics Image Processing*, *27*, 177–194.

McNeill, D. (1982). *Hand and mind : What gestures reveal about thought*. The University of Chicago Press.

Medioni, G. & Nevatia, R. (1984). Matching using linear features. In *IEEE Transactions on Pattern ANal. Mach. Intell.*, volume 6 (pp. 675–685).: ACM.

Pook, P. (1994). Teleassistance: Contextual guidance for autonomous manipulation. In *National Conference on Artificial Intelligence*, volume 2 (pp. 1291–1296). Menlo Park, CA: AAAI.

Price, K. (1986). Hierarchical matching using relaxation. In *Computer Vision Graphics Image*, volume 34 (pp. 66–75).

Sacerdoti, E. D. (1975). The non-linear nature of plans. *Proceedings of IJCAI-75*.

Sacerdoti, E. D. (1977). *A structure for plans and behaviour*. New York: American Elsevier.

Schneider, S. & Cannon, R. (1989). Experimental object-level strategic control with cooperating manipulators point-and-direct telerobotics: Interactive supervisory control at the object-level in unstructured human-machine system environments. In *Proceedings of the ASME Winter Annual Meeting* Stanford, CA.

Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control*. Cambridge, MA: The MIT Press.

Shneiderman, B. (1992). *Designing the user interface : strategies for effective human-computer interaction*. Addison-Wesley.

van der loos, M., Hammel, J., Lees, D., Chang, D., Perkash, I., & Leifer, L. (1990a). Voice controlled robot systems as a quadriplegic programmer's assistant. *In Proceedings of the 13th Annual RESNA Conf. Washington, DC*.

van der loos, M., Hammel, J., Lees, D., Chang, D., & Schwant, D. (1990b). *Design of a vocational assistant robot workstation*. Annual report, Rehabilitation Research and Development Center, Palo Alto VA Medical Center, Palo Alto, CA.

Ware, C. & Jessome, R. (1988). Using the bat: A six dimensional mouse for object placement. *IEEE Computer Graphics and Applications*, *8*(6), 155–160.

Wiemer, D. & Ganapathy, S. G. (1989). A synthetic visual environment with hand gesturing and voice input. *Proc CHI'89*, (pp. 235–240).