Multi-Modal Direction of a Robot by Individuals with a Significant Disability

Shoupu Chen, Zunaid Kazi, Richard Foulds, Daniel Chester
Applied Science and Engineering Laboratories, University of Delaware/A.I. duPont Institute
Wilmington, DE, USA

Introduction

One of the most challenging problems in rehabilitation robotics is the design of an intuitive and efficient interface between the user and the manipulator In general, prototype interfaces have taken two approaches to achieving effective use by individuals with disabilities. Many employ commands which are issued by the user and activate the robot to perform pre-programmed tasks. Others have sought to give the user direct control of the manipulator's motions.

In this project, we propose a new hybrid interface strategy is examined. This new man-machine interface combines command and control approaches to provide for user direction of the robot through the use of multiple modes of interface in conjunction with sophisticated capabilities of the machine. Users of this system use gestures (pointing) to indicate locations, and spoken commands to identify objects and actions. The use of multiple modes of control and command allows the user to operate the robot in a manner which more closely matches the user's needs. The operation is expected to be superior to conventional methods since it capitalizes on the strengths of the user's abilities and coordinates these abilities with software and hardware sophistication of the robot.

This multi-modal approach is based on the assumption that the user's world is unstructured, but that objects within that world are reasonably predictable. Our work reflects this arrangement by providing a means of determining the three-dimensional contours of objects and surfaces which are in the immediate environment (regardless of the actual objects), and an object-oriented knowledge base and planning system which superimposes information about common objects on the three-dimensional world. This paper describes these two components. A third aspect of the project, which will be described in a subsequent paper, involves the user interface which transduces the gesture and identifies the portion of the contour of interest to the user, and the speech recognition interface through which the user accesses the knowledge-base.

Project Overview

The rehabilitation robotics research literature describes many demonstrations of the use of robotic devices by individuals with disabilities. These are reviewed in Foulds [8] and Gilbert and Trefsger [11]. In general, the existing interface strategies have not met the desires of the disabled community. In general, the prototype interfaces have taken two approaches to achieving effective use by individuals with disabilities. Many have commands which are issued by the user and activate the robot to perform pre-programmed tasks. Others have sought to give the user control of the manipulator's motions.

Command Based Interface

The command based interface is one in which the robot is programmed to execute pre-defined movements. The early work on the APL/JHU [24] Robotic Arm Work Station describes a system which uses a mechanical arm under user command. The concept was to place items in fixed locations on the surface of the work-space and to use prestored trajectories and operations to carry out desired functions. Examples of pre-stored functions include:

- picking up a telephone and moving it close to the user's ear
- hanging up the telephone
- eating with spoon in a plate
- eating from a bowl
- turning computer on and off
- moving into position for use

The users entered their selection of a command through a mechanical keypad. In addition to the commands, the system offered modest control of the joints of the robot. A proportional control of the movement of individual joints was possible in order to move the robot to locations that were different from the pre-programmed locations. Similar types of pre-programmed commands were employed in vocational workstations [10, 9]. Considerable work has also been published on the robot workstation designed at Stanford University and the Palo Alto VA Medical Center [12, 26, 27].

Control Based Interface

In contrast to the command oriented rehabilitation robots, there have been a number of projects in which the user directly controls the movements of the manipulator much like a prosthesis. Zeelenberg [28] describes a small robot whose movements are controlled by the position of a track ball. The user has complete control of all movements. None is pre-programmed. Earlier work on the Spartacus Project and the Manus Project [17, 18], shows the potential for multiple degrees of freedom control over the robot. The movements of the elements of the robots are controlled with proportional devices, like joysticks and track balls. This approach offers tremendous flexibility since there are no restrictions for a preset number of commands, a structured environment, or machine *knowledge* of the objects in the world, but is likely to be too demanding of many prospective users. An intelligence with some built-in intelligence is needed to lighten the cognitive and physical load on prospective users.

Integrating Command and Control

The limitations of a command-based interface were discussed Michalowski et al [20] . While modern speech recognizers provide access to large numbers of stored commands, these investigators present the case that effective command of a robot will require use of more commands than is reasonable for the user to remember. As the number of possible commands grows, the human/machine interface becomes increasingly unmanageable. They propose greatly expanding the capability of the robot to not only recognize spoken words, but also understand spoken English sentences. In a continuation of this work, Crangle et al [5] provided an example where the user spoke the sentence, "*Move the red book from the table to the shelf.*" The proposed system would recognize the spoken sentence and understand the meaning of the sentence. The system would have a knowledge of the immediate world so that the robot knew the locations of the table and shelf, as well as the placement of the book on the table. The knowledge base must also know a great deal about the items in the environment. It must understand what books and tables are. It must also know that the book is red as opposed to blue, and red as opposed to read. Additionally, this system must know that the book has not been moved to a different location. While the use of such *natural language* interfaces is extremely interesting, and would offer great benefit, the limitations are many. The requirement that the world be entirely structured so that the robot knows precisely where every item is, is likely to be too demanding. No consumer lives in such a world. In addition, the inclusion of a vision system to accommodate a less structured environment will require the ability to perform object recognition. Both the object recognition and the natural language understanding capabilities required by this effort are the subjects of large scale computer science and artificial intelligence research which is being conducted outside of the rehabilitation field. Until such time as this research progresses to the point where applications are possible, such a sophisticated system is not likely to meet consumer needs. A different approach to command-based robot operation was proposed by Harwin et al [13]. 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 very limited and structured environment. This system did not easily lend itself to a variety of locations and may not be flexible enough to accommodate the needs of individuals with disabilities. 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.

User direction of a Robot

The **Multi-Modal User-Direction Project** addresses the integration of command-based and control-based interfaces to provide an effective user/robot system that can be used in a variety of settings and in both structured and unstructured environments. The project will employ multimodal user inputs including voice recognition for commands and gestures (pointing) for locations (end points) and movement paths (trajectories). This user direction (a combination of command and control) will provide for rapid operation of the manipulator. It will employ the power of predefined commands in conjunction with the flexibility of user control. In keeping with consumer priorities, the robot may be attached to the user's wheel-chair and the user should be free to use the arm in any environment in which he/she travels. No particular structure will be imposed on the environment.

Illustration of the Approach

The effectiveness of this approach is demonstrated in the following example. In the conceptualized scene shown in Figure 1, an individual with a disability uses an electric wheelchair and a portable robot arm. The user wishes to move the pen, which is on the desk, to the box. The user (in this example using a head pointer), points to the pen and says, *move*. The user then points to the box, and says, *there*. The combination of the initial pointing accompanied by the command, *move*, tells the robot to pick up an object at a specific location. The combination of the subsequent pointing and the command, *there*, tells the robot where to move the object.

Put That There

The idea of multimodal user direction is not entirely new, having been discussed extensively by Bolt [3] of the MIT Media Laboratory. Bolt introduced the expression *put that there* in describing his work in optimizing the interface between a user and a large graphical display. The system combined human supervisory control (visual selection and feed-back) with voice commands and gestural pointing. Cannon [4] at Stanford extended this concept to three dimensional robot operation. Cannon's system has worked quite well in laboratory trials. However, it presents problems when being considered as an interface for rehabilitation robotics. The requirement that the user control two video cameras acting as a manually operated rangefinder makes this less than desirable for an individual with disabilities.

Machine Vision System

A brief review

Machine vision has been included in several rehabilitation robotics projects. Harwin and Ginige built a robot-vision system at the Cambridge University, Engineering Department for educating teenagers to perform sorting and building blocks [13]. Komeda and Uchida in the Center of Education and Research, Shibanra Institute of Technology, in Japan, developed a mobile robot system with the vision for bedridden patients to reduce the burden of people who care for those patients [15]. This system consists of a manipulator, a small camera attached to the wrist part of the manipulator and a mobile unit. It gives the surround information to the user through the camera and monitor. Other relevant experimental projects can be found in Detriche & Lesigne [7] and Perala et al [21].

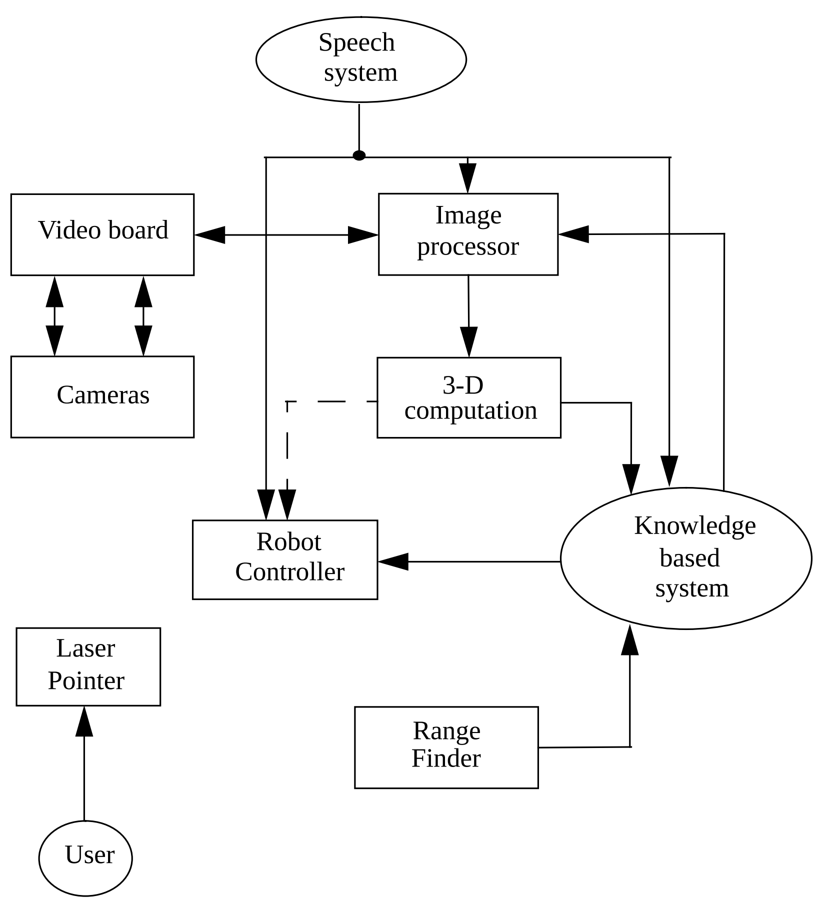

Figure 1. System Configuration

More recently, Cannon [4] at Stanford developed a “Point and Direct” system. Commands are entered via a speech recognizer. The pointing is done in three dimensions. A mobile robot is equipped with two video cameras, each with motorized tilt capabilities. A user views the output of both cameras on two video monitors. The center of view of each camera is marked by a cross-hair. The tilt of the cameras is operated under user control (with a joystick). The pointing for this “put that there” system is accomplished by the user aiming each camera so that the cross-hairs line up on the desired location. The computer calculates the distance from the cameras (robot) to the object by measuring the convergence of the cameras. Bagchi and Kawamura at the Center for Intelligence System of Vanderbilt University adopted the computer vision technique in their Distribute Object-oriented Robotics system for feeding the persons with disabilities. A CCD camera is used to recognize objects on the table and another is used to track the user’s face in a 2-D space.

Vision system design

A key to the success of this system will be the recognition of the location to which the user is pointing. Bolt’s original “put that there” concept calculated the intersection of a line defined by the user’s pointing and the surface of the two dimensional display. Cannon calculated the convergence of two video cameras to calculate a three dimensional distance. This project will use a computer vision system, see Figure 1, and a laser ranging system to determine the complex three dimensional surface which is in front of the user.

The vision systems will employ structured lighting in a technique known as active distance measurement. Rather than having one flat surface (the graphical display) as in Bolt’s studies, this system will have a complex 3-D surface. A table with nothing on it will appear as a large flat, horizontal surface. A table with a pencil and box will be seen as a surface which is essentially flat, with a large lump (the box), and a smaller lump (the pencil). The user will point with a small optical pointer (e.g. a small laser pointer for classroom lectures), as in Figure 2. The vision system will see the light reflected at the point of intersection with the object, and the computer can calculate the location of the point on the three-dimensional surface. This will be translated into coordinates for the robot. The reflected light also provides feedback to the user.

Stereo Vision

The existing stereo vision techniques are classified into several categories:

Full scale nonlinear optimization method—This method is based on an elaborate nonlinear model for imaging and thus allows easy adaptation of any arbitrary accuracy requirement. It needs a computer-intensive full scale nonlinear search and a good initial guess to start the nonlinear search (Abdel and Karara).

Solving linear equation involving perspective projection transformation matrix elements—In this method the coefficients of the 3x4 perspective transformation matrices are regarded as unknown parameters. Given 3-D world coordinates of a number of points and the corresponding 2-D image coordinates, the coefficients in the perspective projection transformation matrices can be solved by the least-squares method. This method does not need nonlinear optimization (Hall and Tio).

Two plane method—This method is used for computing the line of sight ray in global coordinate system given the image coordinates. In this method all of sight are not forced to go through the same point (unique lend center). Given an image point, two points in the global space are computed in two calibrated planes (Martines)

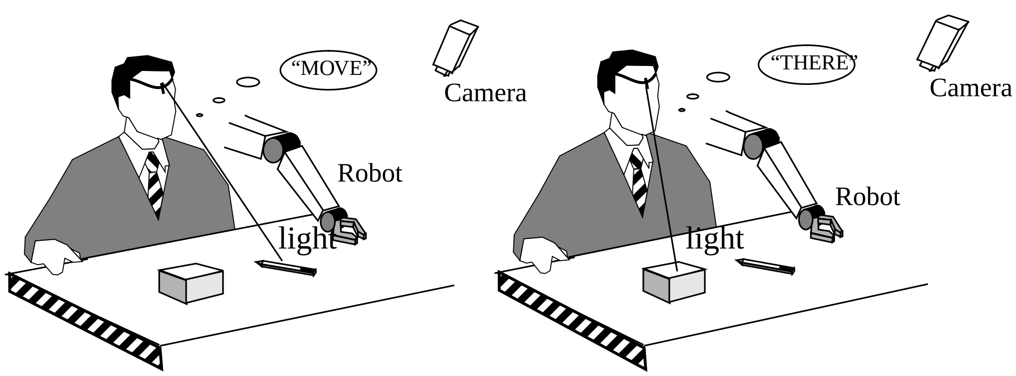

Figure 2. Combination of gesture and spoken command

Geometric method—This method uses geometric construction to derive direct solution for camera location and orientation. Lens distortion and image scale distortion can not be accounted. No nonlinear search is needed. Focal length needs to be given (Matthies).

In this project, we adopt the pinhole camera model and use the straightforward linear least-squares method to solve for the perspective transformation matrices. Two cameras are used in the vision system.

Camera calibration

It is convenient to use a two-coordinate system to describe the geometrical relationship between a three-dimensional object point and its projected image point. The mapping from the global coordinates to the image coordinates can be implemented through the perspective transformation

$$\begin{bmatrix} wx^i \\ wy^i \\ w \end{bmatrix} = \begin{bmatrix} A^i_{11} & A^i_{12} & A^i_{13} & A^i_{14} \\ A^i_{21} & A^i_{22} & A^i_{23} & A^i_{24} \\ A^i_{31} & A^i_{32} & A^i_{33} & A^i_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where x^i and y^i are the 2-D coordinates for the i^{th} image, w is the scale factor, X, Y, and Z are the corresponding 3-D coordinates in the global space. The matrix $\left[A^i_{jk}\right]$ is the perspective transformation matrix for the i^{th} camera. The unknown parameters A^i_{jk} can be estimated by using least-squares optimization method with n $(n > 6)$ noncoplanar known locations, X_n, Y_n, and Z_n in the global space.

3-D location finding

The interested object point coordinates X_o, Y_o, and Z_o in the global space can be easily find based on the inverse mapping from the 2-D image space to the 3-D space using the estimated perspective transformation matrices $\left[A^i_{jk}\right]$, where $1 \le i \le N$, and $N \ge 2$. The location of the object point can be determined as $P = (M^T M)^{-1} M^T N$

where $= \left[X_o \ Y_o \ Z_o\right]^T$, and

$$= \left[A^1_{34}x^1 - A^1_{14} \quad A^1_{34}y^1 - A^1_{24} \quad \dots \quad \dots \quad A^N_{34}x^1 - A^N_{14} \quad A^N_{34}y^N - A^N_{24}\right]^T$$

and

$$= \begin{bmatrix} A^1_{11} - A^1_{31}x^1 & A^1_{12} - A^1_{32}x^1 & A^1_{13} - A^1_{33}x^1 \\ A^1_{21} - A^1_{31}y^1 & A^1_{22} - A^1_{32}y^1 & A^1_{23} - A^1_{33}y^1 \\ \dots & \dots & \dots \quad \dots \quad \dots \\ A^N_{11} - A^N_{31}x^N & A^N_{12} - A^N_{32}x^N & A^N_{13} - A^N_{33}x^N \\ A^N_{21} - A^N_{31}y^N & A^N_{22} - A^N_{32}y^N & A^N_{23} - A^N_{33}y^N \end{bmatrix}$$

It can be seen from the last two sections that the explicit measurement of rotation angles and position of the cameras with respect to the global coordinate system are not required in the computation of the 3-D location of the interested object point.

A primary experiment has been conducted to test the stereovision system. Fifteen posts are mounted on an optical plate with known X, Y, and Z positions as shown in Figure 3a. In general, the stereo vision technique is very accurate in the coordinate frame perpendicular to its cameras optical axis which are roughly parallel to the Z axis of the global coordinate system. Therefore, in Figure 3b the plot shows the Z direction recovery which is the least accurate results among three directions. The median of the absolute error in the Z direction is 1.25%. Figure 3c shows the error of the recovery in Z direction.

The experiment results

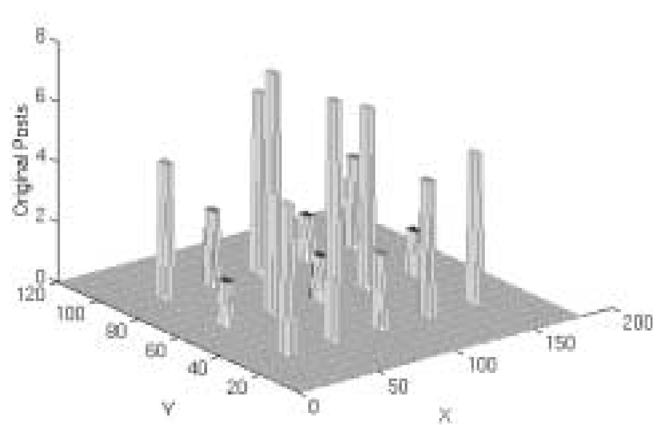

Figure 3a

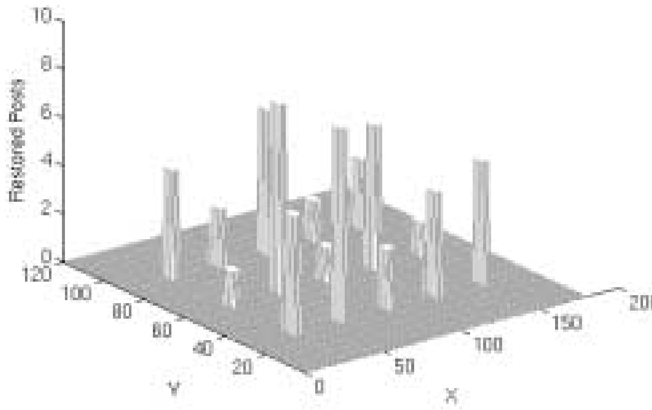

Figure 3b

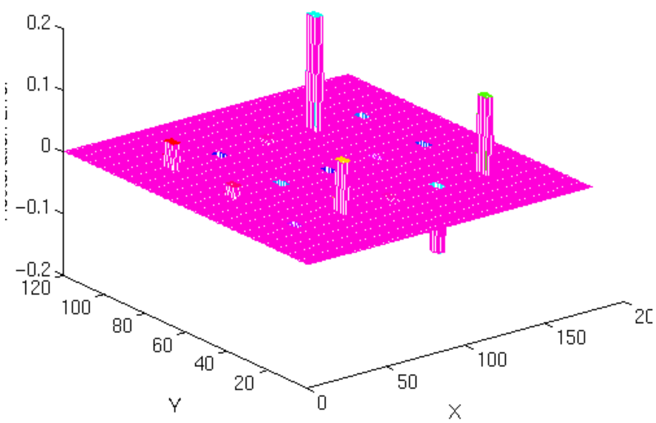

Figure 3c

An object-oriented knowledge-based approach to planning

The Architecture

In order to add objects and actions to the system, we propose an architecture for task planning which is based on an **Object Oriented Knowledge Base**. The knowledge base is essentially composed of two parts:

1. A Knowledge Base of objects

2. A Knowledge Base of actions.

In addition there is a *World Database* where the workspace information is stored and a planner that uses the object knowledge base, the world database, and the action knowledge base to construct robot plans. In order for the whole system to work coherently, we also require a domain theory that contains information regarding temporal and spatial relationships between objects. This domain theory is basically a set of predicates that describe the state of the world and relationships between objects that are pertinent to the domain.

The Object knowledge base

Objects are represented in an increasingly specialized sequence of objects in an inheritance hierarchy. At the top level, we start with a generic abstract object and at the bottom we end up with specific objects whose attributes are fully specified. From the abstract top level objects, we derive objects with intermediate levels of specializations; the choice of these intermediate classes of objects is dependent on the kind of general objects that the system might encounter and the set of tasks that the system might be called on to perform on these objects. 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 the following set of attributes:

Shape—The shape of an object is essential to determine how to grasp and pick up an object. A long and narrow object like a pencil will need a different set of primitive actions than an object that is cylindrical, such as a soda can.

Size/dimensions—The size attribute can be unfilled or it can be expressedly stated, depending on the degree of specialization. For example, at the top level the size has no meaning, while at the bottom level, the size can be exactly stated. At intermediate levels, we need to be able to represent general dimensions. For example, if we have a generic object called a *cup*, we need to be able to specify the general dimensions of a cup, such that the object remains a cup; while if we have a specific object like a specific cup that the user encounters in her world, then the exact dimensions can be properly specified.

Weight—This attribute is again dependent on the degree of specializations. From being unfilled at the top level, the exact weight may be specified for specific objects.

Approach point—Certain objects can only be approached from a specific direction and orientation in order to be grasped.

Grasp point—Certain objects may have specific points where the objects may be grasped. The location of these grasp-points needs to be specified.

Constraints—Constraints may be placed on objects which further constrain approaching, grasping, and moving primitive actions. For example, we may place a constraint on a cup such that the cup is moved in a specific orientation in order to prevent spillage. 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 must be over-ridden if the action involves pouring something out of the cup. Thus the representations of constraints in this field is further qualified by which actions these constraints are applicable to.

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

World Data Base

In addition to the knowledge base of objects, the system also maintains a data base of objects that it sees in the domain, called the *Domain Base*. The objects in the domain contain additional attributes which get instantiated after objects have been identified by the system. Currently, the attributes considered necessary are location and orientation, and attachments to other objects and the workspace.

Example

A very simple example of the object hierarchy is shown below. Prior to interaction with the user, the system sets up the domain-base 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 world data base from the information obtained by the vision system. We are not at this point envisaging the system doing any sort of object recognition. Based on the premise that the user is in the loop, the user points to a *blob* and identifies it to the system. For example, she may point to a specific blob 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 knowledge base. 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 knowledge base which is known as *my-cup* and the blob in its domain-base is replaced by the exact *my-cup* that the system knows, and the attributes of *my-cup* in the domain-base is set up from the knowledge base and information derived from the snapshot of the world. 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 the blob is identified at a certain degree of generalization, such as *long-narrow* etc.

Action Knowledge Base

At this moment we are considering a **STRIPS**-like [22, 23] planning mechanism, where plans have the following general format:

- Action
- Preconditions:
- Constraints:
- Sub-Tasks:
- Goals:
- Effects:

Preconditions must be true before the action can be executed and constraints must remain true during the execution of the action. The top level task may be broken down into more primitive sub tasks, each of which itself is represented as **STRIPS**-like plans. Goals are the primary effect of invoking the action, and effects are the changes that occur in the world as a result of execution of the plan. The main difference between conventional **STRIPS**-like planning and the proposed system is that we take full advantage of the underlying object oriented nature of the objects which drives the planning mechanism. Plans in this model are considered as general templates of actions, where plan parameters are instantiated from the object knowledge base during the planning process. For example, the constraint 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 constraint 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. All plan slots may be instantiated from information obtained from *objects* on which they are invoked in a similar manner.

Another way in which the object oriented paradigm has extended the classical **STRIPS** planning mechanism is illustrated below. 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 that Wilkins identifies. Certain tasks which can be generally handled for most objects may not be applicable to certain objects in the real world. Suppose we have an instrument that is used often in the domain of the user. The instrument has a peculiar shape and must be picked from a specific point. In order to approach the grasp-point, it may not be possible to just simplify specify a certain approach point and assume that the robotic arm will then be able to pick up that instrument. 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* field of an object. In a manner similar to the way action slots are filled depending on the object on which the actions are invoked, subtask slots are also filled, if so specified, from the object's *plan-fragment* field.

This is illustrated with an example as follows: Suppose the generic *Move* action is specified as follows. Move *object to-location* This action may be hierarchically decomposed into sub-tasks, Open (d1) Approach [*object.plan-fragment.approach*][*object at-location*] Close (d2) MoveTo (*to-location*).

The approach sub-task has a default general plan (the general approach action which will be invoked with the parameters specified in the second set of square brackets), as well as an object based plan (specified inside the first set of square brackets). If the object on which this *Move* plan is being invoked does have an approach plan in its plan-fragment field, then that plan is invoked to satisfy the goal. In the case of the example of an instrument given previously, the plan-fragment field will contain an approach sub-slot where a specific plan of approaching the instrument may be fully specified. Thus we see that this integration of knowledge base planing with an object oriented approach allows us to use general plans whenever we can but as well allow us to develop plans for specific objects peculiar to the domain without the need to perform computationally expensive operations. Moreover, each action has a generalized version and specialized versions that are invoked according to knowledge about the object. This allows us to abstract out the general features of an action and invoke them on objects about which the knowledge base might not have any information. It also allows us to view an action as a single action that is applicable to many kinds of objects instead of as a set of actions, each applicable to only one kind of objects as is done in other **STRIPS**-like systems.

Conclusion - A General Illustration

The user approaches a table on which there are a bottle and a glass, both of which are in the knowledge base. The user points to the bottle, and says, *bottle*. From this, the system knows how to approach the bottle. The user points to the glass and says *glass*, *above there*, indicating that the object is a glass, and the final location of the bottle is above the glass.The knowledge base provides additional information. The system, knowing the two height dimensions, computes the distance needed to be *above* the glass. The bottle is moved to its desired destination. The user says, *pour*, which initiates pouring by rotating the bottle in a pouring motion directly over the glass. The user monitors the process and stops the pouring by saying, *stop*. A command, *return*, restores the bottle to its original location. The command, *home*, moves the robot out of the way.

Illustration of the approach (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 the *my-cup* object base, the planner will then calculate the path that needs to be followed to place the *my-cup there*, while maintaining the constraint that the cup must be kept at a certain orientation.

Illustration of the approach (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 the 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.

Illustration of the approach (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*, 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. In the third case, 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. During the process the system as usual updates as much as it is able to the attributes for the mug object. In the fourth case, the user may directly control the arm in approaching and grasping the object. During the operation the system again instantiates the attributes for the object for later reuse.

Acknowledgments

Work on this project is supported by the Rehabilitation Engineering Research Center in Rehabilitation Robotics, NIDRR Grant #H133E30013, and Nemours Research Programs.

References

[1] Adams, L., Gilsbach, J. M., Krybus, W., Meyer-Eberchi, D., Mosges, R., & Schlondorff, G. (1990). CAS-A Navigation Support for Surgery. In *3d imaging in medicine* (pp. 411–423). Springer-Verlag Berlin Heidelberg.

[2] Adjouadi, M. (1992). A Man-Machine Vision Interface for Sensing the Environment. *Journal of Rehabilitation Research and Development*, *29*(2), 57–76.

[3] Bolt, R. A. (1980). “Put-That-There”: Voice and Gesture at the Graphics Interface. *Computer Graphics*, 14(3), 262–270.

[4] Cannon, D. (1992). Point and Direct Telerobotics: Object Level Strategic Supervisory Control in *Unstructured Interactive Human-Machine System*. Unpublished Doctoral Dissertation, Stanford University, Department of Mechanical Engineering.

[5] 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). Montreal

[6] Davies, B. L. & Hibberd, R. D. (1991). A Surgeon Report for Prostatectomies. *Proc. Fifth International Conference on Advanced Robotics*, (pp. 871–875).

[7] Detriche, B. & Lesigne, B. (1991). Man Machine Interface and Programming Aspects of the Robotized System ‘MASTER’ for Disabled People. *Medical & Biological Engineering & Computing*, *29*.

[8] Foulds, R. A. (1986). *Interactive Robotics Aids–One Option for Independent Living: an International Perspective*, volume Mono graph 37. World Rehabilitation Fund.

[9] Fu, C. (1986). An Independent Vocational Workstation for a Quadriplegic. In R. Foulds (Ed.), *Interactive Robotic Aids-One Option for Independent Living: An International Perspective*, volume Monograph 37 (pp. 42). World Rehabilitation Fund.

[10] Gilbert, M. & Foulds, R. A. (1987). Robotics at the Tuffs New England Medical Center. In *Proceedings of the 10th Annual Conference on Rehabilitation Engineering* (pp. 778–780). San Jose.

[11] Gilbert, M. & Trefsger, J. (1990). In *Proceedings of the 1990 International Conference on Rehabilitation Robotics* Wilmington, DE.

[12] Hammel, J., Van der loose, M., & Perkash, I. (1991). Evaluation of DeVAR-IV with a Quadriplegic Employee. In *Annual Report of the Rehabilitation Research and Development Center* (pp. 99–100). Palo Alto, CA: Palo Alto VA Medical Center.

[13] 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 Robotics 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.

[14] Kanade, T. (1987). *Three-Dimensional Machine Vision*. Kluwer Academic Publishers.

[15] Komeda, T., Uchida, T., Miyagi, M., Koyama, H., & Funakubo, H. (1991). Development of the Mobile Robot System with Visual Sensor for Bedridden Patients. *Medical & Biological Engineering & Computing*, *29*.

[16] Krueger, M. W. (1991). *Artificial Reality*. Addison-Wesley.

[17] Kwee, H. (1986). Spartacus and Manus: Telethesis Developments in France and the Netherlands. In R. Foulds (Ed.), *Interactive Robotic Aids-One Option for Independent Living: An International Perspective*, volume Monograph 37 (pp. 7–17). World Rehabilitation Fund.

[18] Kwee, H., Thonninsen, M., Cremers, G., Duimel, J., & Westgeest, R. (1992). Configuring the Manus System. In *Proc. of RESNA International'92* (pp. 584–587). Washington, DC: RESNA Press.

[19] Masaki, I. (1992). *Vision-Based Vehicle Guidance*. Springer-Verlag.

[20] Michalowski, S., Crangle, C., & Liang, L. (1987). Experimental Study of a Natural Language Interface to an Instructable Robotic Aid for the Severely Disabled. In *Proc. of the 10th Annual Conf. on Rehabilitation Technology* (pp. 466–467). Washington, DC: RESNA Press.

[21] Perala, J., Lounela, L., Ekberg, J., & Lappalainen, V. (1991). Videotelephone-Based Support Service System for Elderly and Disabled People. *Medical & Biological Engineering & Computing*, *29*.

[22] Sacerdoti, E. D. (1975), The Non-Linear Nature of Plans, In *Proceedings of IJCAI-75*.

[23] Sacerdoti, E. D. (1977), A Structure for Plans and Behavior, American Elsevier, NY

[24] APL/JHU robotic arm workstation. In R. Foulds (Ed.), *Interactive Robotic Aids-One Option for Independent Living: An International Perspective*, volume Monograph 37 (pp. 51). World Rehabilitation Fund.

[25] Taylor, R. H., Paul, H. A., Mettelstadt, B. D., Hanson, W., Kanzanzides, P., Zuhars, J. F., Glassman, E., Musits, B. L., & Williamson, W. (1990). An Image-Based Robotic System for Hip Replacement Surgery. *Journal of the Robotics Society of Japan*, (pp. 111–116).

[26] Van der loos, M., Hammel, J., Lees, D., Chang, D., Perkash, I., & Leifer, L. (1990). Voice Controlled Robot Systems as a Quadriplegic Programmer's Assistant. In *Proceedings of the 13th Annual RESNA Conf.* Washington, DC.

[27] Van der loos, M., Hammel, J., Lees, D., Chang, D., & Schwant, D. (1991). Design of a Vocational Assistant Robot Workstation. In *Annual Report of the Rehabilitation Research and Development Center* (pp. 97–98). Palo Alto, CA: Palo Alto VA Medical Center.

[28] Zeelenberg, A. P. (1986). Domestic Use of a Training Robot-Manipulator by Children with Muscular Dystrophy. In R. Foulds (Ed.), *Interactive Robotic Aids-One Option for Independent Living: An International Perspective*, volume Monograph 37 (pp. 29–33). World Rehabilitation Fund.

Address

Shoupu Chen
Applied Science & Engineering Laboratories
University of Delaware/A.I. duPont Institute
P.O. Box 269, 1600 Rockland Rd.
Wilmington, DE 19899
(302)651-6830
email: chen@asel.udel.edu