Gesture-Speech Based HMI for a Rehabilitation Robot

Shoupu Chen, Zunaid Kazi, Matthew Beitler,
Marcos Salganicoff, Daniel Chester, and Richard Foulds
Center for Applied Science and Engineering
University of Delaware / A.I. duPont Institute
Wilmington, DE 19899

Abstract - One of the most challenging problems in rehabilitation robotics is the design of an efficient Human-Machine Interface (HMI) allowing the user with a disability considerable freedom and flexibility. A multimodal user direction approach combining command and control methods is a very promising way to achieve this goal. This multimodal design is motivated by the idea of minimizing the user's burden of operating a robot manipulator while utilizing the user's intelligence and available mobilities. With this design, the user with a physical disability simply uses gesture (pointing with a laser pointer) to indicate a location or a desired object and uses speech to activate the system. Recognition of the spoken input is also used to supplant the need for general purpose object recognition between different objects and to perform the critical function of disambiguation. The robot system is designed to operate in an unstructured environment containing objects that are reasonably predictable. A novel reactive planning mechanism, of which the user is an active integral component, in conjunction with a stereo-vision system and an object-oriented knowledge base, provides the robot system with the 3D information of the surrounding world as well as the motion strategies.

Introduction

Researchers in the Rehabilitation Robotics community have been developing robot systems which can help persons with disabilities to gain access to a universe previously inaccessible to them. 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 [1].

The limitations of a command-based interface were discussed by Michalowski *et al* [2]. The effectiveness of the command systems are limited by the need for a reasonably structured environment and the limited number of commands. In control-based methods, the physical limitations of the user require that the input system be limited to a few degrees of freedom. A conventional 2D joystick is insufficient to adequately control a manipulator. At the other extreme of robot control are the completely autonomous systems that perform with effectively no user supervision, the long elusive goal of AI, 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, strategies of exploration, insufficient sensory information, inherent inaccuracy in the robotic devices and the mode of operation.

Researchers have proposed a number of systems which investigate alternate modes of human-computer interaction in addition to speech and vision based ones. It has long been acknowledged that pointing or deictic gestures are an important part of human-human communication and efforts have in turn been carried out in using gestures and hand pointing as a mode of human-machine interface [3]. Work is also being done in attempting to extend this concept by using multiple modes of human-machine interfacing. The concept of multimodal interfacing has been discussed extensively by Richard Bolt of the MIT Media Laboratory [4]. Bolt introduced the expression “put that there” in describing his work in optimizing the interface between a user and a large 2-D graphical display. Cannon at Stanford extended this concept to three dimensional robot operation [5]. Cannon's system has worked quite well in laboratory trials. However, it presents problems when being considered as a general interface for assistive robotics. The requirement that the user control two video cameras acting as a manually operated range-finder makes it less than desirable for an individual with disabilities.

The concept of multimodal direction of a rehabilitation robot was re-introduced by Foulds in 1993 with a new strategy[6]. This new strategy extends the combined deictic gesture and spoken word of Bolt to true 3-

D environments manipulated by a robot. It details an intuitive and efficient interface between the user and the manipulator as well as a reactive planning mechanism[8]. Users of this system use deictic gestures (pointing, achieved by a head mounted laser pointer) to indicate locations, and spoken commands to identify objects and specific actions. It combines command and control approaches to provide for user direction of the assistive robot through the use of multiple modes of interface which includes voice recognition for commands and gesture (pointing) for locations (end points). This strategy provides for rapid operation of the manipulator by employing the power of predefined commands in conjunction with the flexibility of user control. Unlike other control methods there is no need for the user to operate a joystick or any sort of mechanical devices. With a laser pointer attached to his/her head, the user simply points the laser beam to an object or a location by positioning his/her head in an appropriate way. By using a speech system the user is able to activate different operations of the robot system based on his/her verbal commands. The combination of spoken language along with deictic gestures 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 is used to supplant the need for a general purpose object recognition module in the system. Instead, 3-D shape information is augmented by the user's spoken word which may also invoke the appropriate object properties from the object-oriented representation. In this paper the reactive planner and the vision system are discussed in detail.

SYSTEM DESCRIPTION

The configuration of the interface system is depicted in Figure 1. The backbone computing machine for the vision interface is an SGI XS-24 IRIS Indigo computer. Pictures are taken by two CCD color cameras. Each camera is equipped with motorized TV zoom lens. Cameras are connected to the SGI Galileo image acquisition 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 (Noesis Vision) 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 (Dragon Systems, Inc.) running on a PC. A six degree of freedom robot manipulator, Zebra ZERO (Integrated Motions, Inc.), is employed as the manipulation tool. The planner executes on a Sun Sparc 5. To provide the user with the preview of the plans there is a simulated environment (for details see [7]) on an IRIS INDIGO ELAN SGI machine. Communications between the planner and the subsystems are supported by RPC (Remote Procedure Call) routines, which allow the computers to make procedure calls to each other.

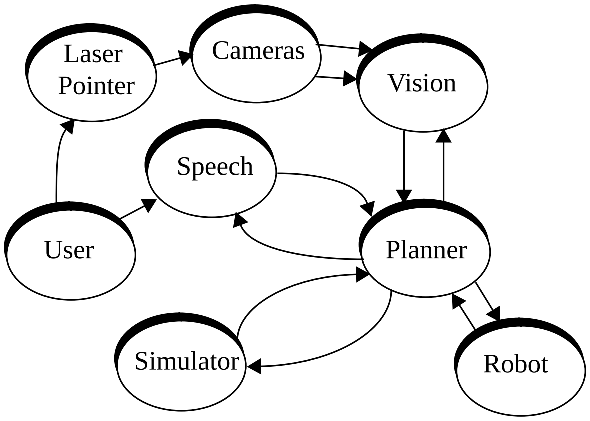

FIGURE 1.System configuration

REACTIVE PLANNER

The planner discussed in this paper proposes a novel reactive planning approach 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. There are assumptions essential for the mechanism to be practical in the real world and to facilitate human-computer interaction as a means of providing reactive and re-planning 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 for reacting to environment changes. Sensing uncertainty and computational complexity precludes the feasibility of a totally autonomous sensor-based reactive planning system, and hence user input is also used to impart a reactive component to the system. The hierarchical human-machine interface and object oriented representation allow 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. Generic plans and specialized plans are supplemented by user interaction. Whenever incomplete information precludes the development of correct plans, the user can rectify this 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. Given this underlying architecture, the system first determines what the user wants, and then makes plans to accomplish the task. When confronted by 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. 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.

WorldBase

Objects are represented in an increasingly specialized sequence of object classes in an inheritance hierarchy. A four tiered hierarchy has been devised, where object classes become increasingly specialized from the top level to the bottom of the hierarchy. At the top level, it starts with a generic abstract object which causes generic plans to be developed for objects about which there is no exact information. The second level object classes are classed in terms of general shapes such as cylindrical, flat, spherical, etc. This enables the planner to modify approach and grasp plans when more information is available for 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 it ends 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 the user 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, the user may place a constraint on a cup such that the cup's orientation cannot be changed during transport to prevent spillage. However, the 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 must be over-ridden if the action involves pouring something out of the cup.

Plan Fragments: Another needed component are the plan fragments that are 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.

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, and attachment relationships to other objects and the workspace.

Object hierarchy 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. *There is no intention to do any object recognition.* Based on the premise that the user is in the planning 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 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* and the *blob* in the DomainBase is replaced by the exact *my-cup* that the system knows, 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 the *blob* is identified at a certain degree of generalization.

PlanBase

The plan knowledge base, PlanBase, is a collection of STRIPS-like plans,[9] and the planner is based on a modified STRIPS-like planning mechanism. The main difference between conventional STRIPS-like planning and the proposed system is that the latter takes 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 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.

Handling exceptions: Another way in which the object oriented paradigm has extended the classical STRIPS planning mechanism is illustrated below. 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 which can be handled generally for most objects may not be applicable to certain objects in the real world. Suppose there is 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-fragment* slot.

Plan adaptation

The planner system can adapt to the changing environment during the course of robot manipulating. Suppose that there is a *mug* in the robot working environment. The user points to the 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 the following:

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 generalization 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.

VISION SYSTEM

In our system an obvious task is to acquire the 3D information about objects of interest as well as locations with respect to the robot operating coordinate system in order that the robot can generate motions accordingly. To accomplish the task, 3D machine vision technology was chosen to provide the three-dimensional contours of objects and the three-dimensional locations of end points. The present paper discusses the use of stereo vision with the help of a structured light (light-point, light-stripes) source to acquire the information about the immediate robot workspace environment. The advantages of using structured lighting are two fold. First, it helps in differentiating object from object and objects from backgrounds based on structure discontinuity in the images. Secondly, the stereo matching algorithm based on structured lighting does not require pixel by pixel operation; thus, it reduces the computational complexity by an order of magnitude compared to correlation-type match algorithms. On another note, to actively find a single laser beam spot in the image plane is quite simple. It can be done by just taking one picture with the laser pointer off and a second picture immediately with the laser pointer on then subtracting the first image from the second one. An adaptive thresholding scheme is further developed to achieve the goal of robustness in extracting the laser spot from the background noise. The method of coordinate transfer from 2D image space to 3D robot space is designed so that it does not require any knowledge of camera parameters such as focal length, position and orientation. All the necessary parameters of the camera are embedded in a single transformation matrix that is estimated through the calibration process. Computing procedures for the transformation matrix are described in [10].

Line-segment pair match

The objective of stereo vision is to recover 3D information about the object in the work environment using images taken from different viewpoints, that is, one camera moved from place to place or multiple cameras fixed in different locations. The most essential and most difficult procedure in stereo vision is image matching. The purpose of the match process is to find the correspondence among the features extracted from two or more images. The difficulty of the image match problem stems from the factors such as image variations due to different perspective projection, the source of lighting, etc. Researchers in this field have been developing various algorithms in past 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 [11]. Early representative works can be found in [12,13,14] in which they use the relaxation labeling technique to solve the stereo image matching problem. More recent developments incorporate structure information between image entities in addition to entity properties to solve the correspondence problem [15, 16, 17]. It should be noted that there is no unified approach to the stereo correspondence problem. In practice, it is very much application dependent.

For the gesture-speech based HMI system the requirement of the vision system is to provide contour information of the objects in the immediate environment. A feature-based matching algorithm is suitable for this application. To reduce the false extraction rate, a high intensity structured-light source with a pattern such as parallel stripes is employed in this design. The distorted light patterns in the images can be 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. In this paper a straight line pattern is selected as it naturally incorporates the figural continuity constraint. A line-segment pair-match scheme is developed based on the geometric characteristics of the features obtained from the images.

Geometric constraints: The stereo cameras are located and oriented so that there is only a horizontal displacement between them. The distance from the focal point of one camera to that of the other is called baseline, and denoted as B. The directions of the optical axes of the two cameras are identical and perpendicular to the baseline. The B value is usually small compared with the scene depth. Further, the focal lengths of the cameras are assumed to be one. Suppose that a global point $u = \begin{bmatrix} u_X & u_Y & u_Z \end{bmatrix}$ is projected onto one of the two image planes, say the left image. Then the lines connecting u and the two focal points determine a unique epipolar plane. The projection of the same point u in the right image must be on the intersection of the epipolar plane and the right image plane. The intersection plane and the image plane is called the epipolar line. Every point on a given epipolar line in one image must correspond to a point on the corresponding epipolar line in the other image.

Let the perspective projection of the 3D point u onto the image be $v^n = \begin{bmatrix} v_x^n & v_y^n \end{bmatrix}$. Define the vector disparity between the two images as $\delta = \begin{bmatrix} \delta_x & \delta_y \end{bmatrix} = \begin{bmatrix} v_x^l - v_x^r & v_y^l - v_y^r \end{bmatrix}$, where the superscripts l and r signify the left and right camera respectively. It can be shown that $|\delta_x| = B/u_Z$ and $|\delta_y| = 0$ under the camera configuration stated above. (In general, $\delta_x, \delta_y \in [-I, \ldots, I]$ where I is an integer, in terms of the number of pixels.) It also can be seen that the magnitude of the disparity is a function $\Re(\ )$ of the depth of the scene u_Z in the cameras' field of view; that is, $|\delta| = \Re(u_Z)$ and obviously $|\delta| \le \Re(u_{Z_{min}})$, where $u_{Z_{min}}$ is the shortest Z distance of 3D points within the cameras' field of view to the cameras.

As discussed before, stripe structured light is used in this study. The structured light projector is placed and oriented so that the results of the perspective projection of the light stripes in the scene are nearly vertical line segments (with very small positive and negative angles with respect to the vertical line for two lines next to each other) in the images. Express the end points of a light stripe in the global space as u^1 and u^2 and their perspective projections as ${}^1v^l$ and ${}^2v^l$, and ${}^1v^r$ and ${}^2v^r$ in the left and right image respectively. As defined in [18] the epipolar constraints for line segment match are $\left|\delta_y^1\right| = 0$ and $\left|\delta_y^2\right| = 0$; the x-disparity constraints for line segment matching are $\left|\delta_x^1\right| < d$ and $\left|\delta_x^2\right| < d$, where $d \le \Re(u_{Z_{min}})$.

Line segment pair match scheme: The epipolar constraints and the disparity constraints stated previously are now used to develop a simple scheme for line segment pair matching in two images. First, denote the line segment sets in two images by $\boldsymbol{L}^l = \{L_s^l\}$ and $\boldsymbol{L}^r = \{L_t^r\}$ where $s = [1, 2, \ldots, S]$ and $t = [1, 2, \ldots, T]$ are the number of line segments in the left and right images. The pair search process takes place between $\{L_s^l\}, \forall s$ and $\{L_t^r\}, \forall t$. A total of $S \times T$ pairs are compared.

For each potential match pair, three penalty measures are defined: $P_\lambda(s, t) = \left|{}^s v_\lambda^l - {}^t v_\lambda^r\right|, \forall s, \forall t$,where $\lambda = [1, 2, 3]$, $P_1(\ )$ stands for x_position_penalty, $P_2(\ )$ for y_position_penalty, and $P_3(\ )$ for length_penalty. For each λ, $P_\lambda(\ )$ is normalized to $\bar{P}_\lambda(\ )$ so that $\bar{P}_\lambda(\ )_{max} = 1$. A total_penalty function for each pair is defined as $P(s, t) = \sum_\lambda \alpha_\lambda \bar{P}_\lambda(s, t), \forall s, \forall t$. The weights α_λ are selected according to the importance of each penalty function in the search process. Unfortunately, there is no general rule for picking the values for these coefficients. (In the experiment of this study, $\alpha_1 = 0.3$, $\alpha_2 = 0.3$, and $\alpha_3 = 0.4$)

The initial set of line segment pairs is determined by searching for the minimum total_penalty $P(s, t)_{min}$ over all t for each s. It is very likely that the initial set does not satisfy the rule of at most one to one mapping and the competition exists for best matching. An adjustment process is thus needed. We describe the adjustment process by an example. Suppose pairs ($s = 5$, $t = 5$) and ($s = 6$, $t = 5$) are the best mappings for $s = 5$ and $s = 6$ after the matching process. By looking at the total_penalty $P(s, t)$ it is found that the second best mappings for s=5 and s=6 are ($s = 5$, $t = 6$) and ($s = 6$, $t = 6$). Examining the numerical values (listed as the elements of a 2×2 matrix):

$$\begin{bmatrix} P(5, 5) = 1.622 & P(6, 5) = 4.954 \\ P(5, 6) = 14.45 & P(6, 6) = 8.007 \end{bmatrix}$$

the highest penalty value $P(5, 6) = 14.45$ eliminates the possibility of accepting the mapping ($s = 5$, $t = 6$). Thus, the probability of mapping ($s = 5$, $t = 5$) is enhanced, which in turn weakens the probability of mapping ($s = 6$, $t = 5$) and enhance the probability of mapping ($s = 6$, $t = 6$).

Post-matching processing and 3D segmentation

The matching process described in the previous section ends up generating W pairs, $W \le min(T, S)$, of line segments from $\{L_s^l\}, \forall s$, and $\{L_t^r\}, \forall t$. That is, $L_i^l \leftrightarrow L_i^r$ where " $\leftrightarrow$ " means "match", $i = [1, \ldots, W]$. It is clear that the matching process just simply discards those "extra" segments found in the larger set of line segments, and unmatchable lines. Upon finding the matched pairs, the 3D recovery is straightforward by applying the algorithm stated in [10] to the matched segments L_i^l and L_i^r. It is important, however, to note that the length of the matched vector L_i^l in the left image does not necessarily equal that of the vector L_i^r in the right image because of occlusion or optical non-linearity. Methods have been developed to tackle this problem but in this research we just use truncation, in other words, only C_i, where $C_i = min[Length\ of\ L_i^l,\ \ Length\ of\ L_i^r]$, expressed as the number of pixels in the smaller line segment, are used in 3D computation for the i^{th} vector pair. It should be pointed out that the number of pixels in the image produced by the video board is 640x486 = 311,040. By projecting light-stripes onto the objects we only process those bright lines in the image. The total number of pixels involved in the process is roughly equal to $C = \sum_{i=1}^{W} C_i$ which is, in general, much less than 311,040. For example, if $W = 50$, $max(C_i) = 200$, then $C \le 10,000 \ll 311,040$.

Now, we have the 3D data set $\{[X_j Y_j Z_j]\}$, where $j = 1, \ldots, C$, describing the shape of the objects with a series of 3D line segments, L_i , $i = 1, \ldots, W$. For each 3D line segment, L_i , let $E_i^{\pm} = [X_j Y_j \pm Z_j]$, where $[X_j Y_j \pm Z_j]$ are the end points of L_i . '+' indicates the end having smaller Y value, and '-' indicates the end having larger Y value. We then apply the clustering process to the points $E_i^{\pm}$ with a Euclidean measure. The clustering process results in segmenting the $E_i^{\pm}$ into different groups. The points $E_i^{\pm}$ are also used to compute the angle of each object with respect to the X axis of the world coordinate system.

EXPERIMENTS

The experiments conducted are mainly used to demonstrate the feasibility of using the Gesture-Speech interface to enable the user to control a rehabilitation robot manipulator in picking and placing objects in the multimodal HMI system. Figure 2 (upper half) shows the scene of two items (one pocket-handbook and one small box) placed on the table; next to those two items is the robot manipulator that is not shown in the picture.

To activate the system, the user issues the following instructions for the robot: (the words in square brackets imply speech combined with a pointing gesture)

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

A brief explanation of the plan follows.

put - A ***TASK*** specification; Semantic analog to a Verb in Natural Language.

[that]- Deictic that gets instantiated to a ***THING*** (a book in this case). 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)***

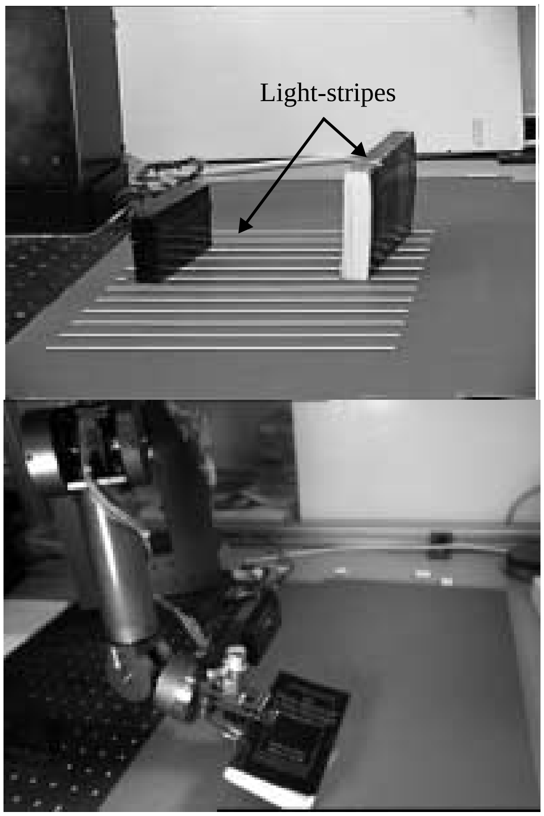

FIGURE 2.Put that there

Another example is ‘drinking’. The task is to pick up the straw, put the straw into the paper cup, move the cup towards the user (see Figure 3). The instructions can be

```
Put this [straw] in this [cup];

move cup [here].
```

In the first instruction the components are

put-in:->***TASK***

*straw:->****THING (TASK-FOCUS)***

*cup:->****LOCATION (DESTINATION)***

In the second instruction the location of the *cup* is already known so there is no need of a pointing gesture. The components again are

move:->***TASK***

*cup:->****THING (TASK-FOCUS)***

*here:->****LOCATION (DESTINATION)***

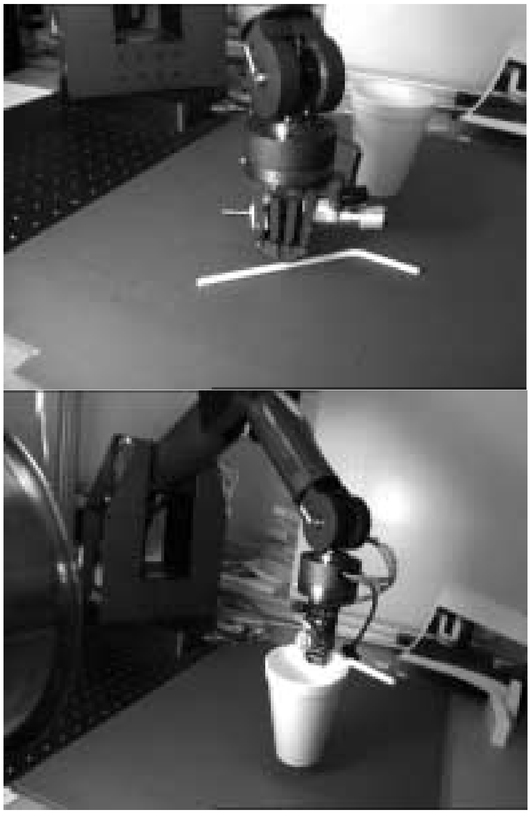

FIGURE 3.‘Drinking’ example

CONCLUSIONS

In this paper we have presented a novel human-machine interface system for persons with physical disabilities to operate an assistive robot arm. Equipped with vision and speech technologies, this system incorporates gesture-speech human actions with a reactive task planning mechanism so that it maximizes the utilization of human intelligence while it minimizes the user's burden of operating a robot. Our goal is to achieve 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 intrinsic 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 loading of the system is decreased, the system becomes more flexible, pleasant to use and less fatiguing, and is ultimately a more effective assistant.

ACKNOWLEDGEMENT

Support for this research has been provided by the Rehabilitation Engineering Research Center in Rehabilitation Robotics, NIDRR Grant #H133E30013, and Nemours Research Programs.

REFFERENCES

[1] Foulds RA. *Interactive Robotics Aids–One Option for Independent Living: an International Perspective,* volume Mono graph 37. World Rehabilitation Fund, 1986.

[2] Michalowski S, Crangle C, Liang L. "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*. Washington, DC: RESNA Press, 1987;466–467.

[3] Wiemer, D., & Ganapathy, S. G. "A synthetic visual environment with hand gesturing and voice input," *Proc CHI'89*, pp 235-240, 1989.

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

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

[6] Chen S, Kazi Z, Foulds R, Chester D. "Multimodal direction of a robot by individuals with a significant disability." In: *Proceedings of the 4th International Conference on Rehabilitation Robotics*. Wilmington, DE, USA, June 1994.

[7] Beitler M., Foulds R., Kazi Z., Chester D., Chen S., and Salganicoff M., "Multimodal User Supervised Interface and Intelligent Control (MUSIIC)," *Proc., AAAI 1995 Fall Symposium on Embodied Language and Action,* Cambridge, MA.

[8] Kazi Z., Beitler M., Salganicoff M, Chen S., Chester D., and Foulds R., "Intelligent telerobotic assistant for people with disabilities," *SPIE, Telemanipulator and Telepresence Technologies II, 1995.*

[9] Sacerdoti, E. D. *A Structure for Plans and Behavior*, American Elsevier, NY, 1977.

[10] Chen S., "Development of an Active Three-Dimensional Vision Interface for a Rehabilitation Robot System," *Technical Report ROB9505, ASEL, University of Delaware/A.I. duPont Institute, Wilmington, DE,* December 1995.

[11] Barnard DT, Fischler MA., "Computational stereo." *ACM Computing Survey* 1982;14(4):553–572.

[12] Barnard ST, Thompson WB., "Disparity analysis of images." *IEEE Trans. on Pattern Anal. Mach. Intell.* 1980;2(4):333–340.

[13] Medioni G, Nevatia R. "Matching images using linear features." *IEEE Trans. on Pattern Analy. Mach. Intell.* 1984;6(6):675–685.

[14] Price K. "Hierarchical matching using relaxation." *Computer Vision Graphics Image Proc.* 1986;34:66–75.

[15] Boyer KL, Kak AC. "Structural stereopsis for 3d vision." *IEEE Trans. on Pattern Anal. Mach. Intell.* 1988;10(2):144–166.

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

[17] Matsuyama T, Arita H, Nagao M. "Structural matching of line drawing using the geometric relationship between line segments." *Computer Vision Image Proc.* 1984;27:177–194.

[18] Chou S, Tsai W. "Line segment matching for 3d /computer vision using a new iteration scheme." *Machine Vision and Applications* 1993; 6:191–205.