An Egocentric Vision based Assistive Co-robot

Jingzhe Zhang[#1], Lishuo Zhuang[§1], Yang Wang[§1], Yameng Zhou[#1], Yan Meng[#], Gang Hua[§]

[#]Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ 07030
{jzhang23, yzhou5, yan.meng}@stevens.edu

[§]Department of Computer Science
Stevens Institute of Technology
Hoboken, NJ 07030
{ywang7,lzhuang,ghua}@stevens.edu

Abstract—**We present the prototype of an egocentric vision based assistive co-robot system. In this co-robot system, the user is wearing a pair of glasses with a forward looking camera, and is actively engaged in the control loop of the robot in navigational tasks. The egocentric vision glasses serve for two purposes. First, it serves as a source of visual input to request the robot to find a certain object in the environment. Second, the motion patterns computed from the egocentric video associated with a specific set of head movements are exploited to guide the robot to find the object. These are especially helpful for quadriplegic individuals who do not have needed hand functionality for interaction and control with other modalities (e.g., joystick). In our co-robot system, when the robot does not fulfill the object finding task in a pre-specified time window, it would actively solicit user controls for guidance. Then the users can use the egocentric vision based gesture interface to orient the robot towards the direction of the object. After that the robot will automatically navigate towards the object until it finds it. Our experiments validated the efficacy of the closed-loop design to engage the human in the loop.**

Keywords—co-robot, active learning, human in the loop

I. INTRODUCTION

[1]Robotic technology has the potential to offer considerable help for those disabled individuals. Rehabilitation and assistive robotic systems have been developed to assist elders [1] [2] and especially the disabled ones [3] to improve their quality of life (QOF). Handy 1 Automated Eating Aid [4] is designed to enable people with severe disabilities, such as those quadriplegic individuals, for regaining independence in important daily living activities, such as eating, drinking, and washing, etc. Care-O-bot II is another example of the assistive robotic system [5] to help those elderly and handicapped people by providing a touch screen to control the robot. However, for quadriplegic individuals who lost their hand functionality, it is very important to explore other modalities to design a human robot interface that they actually operate to control such assistive robotic systems.

We present a prototype co-robot system that is equipped with a pair of egocentric video glasses, which captures the visual scene from the first person view (FPV). The intended application of this co-robot system is to help a person with severe disability for grabbing a target object. Since grabbing by itself is a very intricate task, in this paper, our prototype co-robot system focused on the task of object finding, with the user in the loop. In this co-robot system, we exploit the egocentric camera built on the glasses to analyze the head movement of the users to guide the robot finding the target object, which provides a very convenient user interface for those people who are unable to operate the co-robot by their hands. Since the design goal of our system is to assist those severely disabled, such as quadriplegic individuals, the head gestures used for control has to be natural and easy to perform. Therefore, in order to provide a comfortable user experience, only simple and natural head gestures such as nodding, shaking head and turning head around have been considered to be included in the gesture lexicon to trigger interactive commands in our human-robot interaction system. The user's gestures are estimated by motion-pattern estimation with vision based motion sensing similar to our previous work [6].

To facilitate the object finding task, it is essential to have a real-time visual object recognition system. The user looks at the target object and nods his/her head, this will trigger the system to take a picture of the target object and simultaneously send a command to the co-robot through a central server to find that object. If the robot cannot find the object within a pre-specified time period, it will ask for the user's help through speech, *i.e.*, by saying "I need help". In our prototype system, this is achieved by text-to-speech and executed in the central server. Then the user would use the head motion gesture control interface to guide the robot towards the direction of the object and then using the nod gesture to issue a "continuing search in that direction" command. The robot then automatically navigates towards the object, avoiding any obstacles during the navigating process. For object recognition, we exploited the vocabulary tree algorithm proposed by Nister and Stewenius [7] with a post verification scheme using Homography.

The rest of the paper is organized as follows. Section II presents some closely related existing assistive robotic systems. Section III is dedicated to the proposed system framework of the co-robot system. Section IV describes the details of the algorithms and techniques we employed as well as some improvements. Section V presents the experimental results to evaluate the performance of several major technical components in the system, as well as the overall system performance. Finally we conclude our paper in Section VI.

II. RELATED WORK

Assistive robotic systems have been developed rapidly, ever since early 1990's. For instance, a rehabilitation robot called

[1] The first four authors made equal contributions to the paper.

978-1-4673-6024-1/13/$31.00 ©2013 IEEE

Handy 1 [4] (Fig. 1) used a scanning system with seven lights and a user controlled switch to help people choosing the dishes of food that they want to eat. It also provided different functionalities such as drinking, washing, shaving, teeth cleaning and even applying facial make-ups. At that time, the simple interaction design, in a large sense, alleviated the difficulty of disabled people in their normal life.

Fig. 1: In Clockwise A: Handy 1 System, B: The cup attachment, C: Washing, Shaving and Teeth Cleaning Tray D: Handy 1 Eating tray section.

According to the concept of rehabilitation robots such as Handy 1, Takahashi *et.al*. have developed the assistive robotic system (Fig. 2) with a head pointing device and an emotion evaluation sensor [8] to find objects and give feedback, respectively. Users can easily control the robot feeding them by their head actions. This feature had greatly minimized the unnecessary effort for user to trigger related robot actions, and the emotion evaluation could give the feedback to the robot whether it acts properly or not.

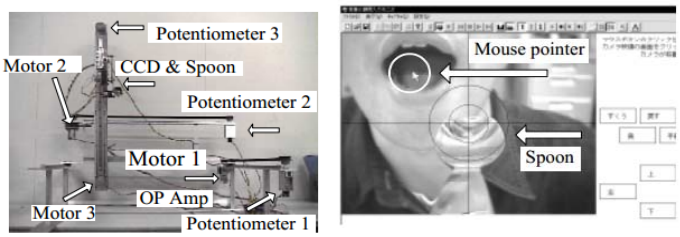

Fig. 2: Left to Right: (a) the Robot system; (b) The PC display during operation using the CCD camera.

Handy 1 [4] and Takahashi's robot [8] are both eating assistant robotic systems fixed on a table, and only working in certain known environments. There are different types of assistive robotic systems that have been presented in [9]. Care-O-bot II [5] (Fig. 3) provides a standalone robot system to assist disabled people by using recognitions and control techniques. Users can control Care-O-bot II by two industrial PCs and a hand-held control panel. It can aid users to their desired destination and meanwhile avoid obstacles. With some additional sensors in the robot's hand the Care-O-bot II could grasp any suitable objects.

Other assistive robots presented laswer pointers [10] [11] as a user interface. However, all kinds of interface require human hands to operate which are significantly inconvenient for quadriplegic individuals.

Fig. 3: Left to right: (a)Grab functionality; (b) The walking aid system.

III. System Framework

Our egocentric vision based assistive co-robot system has three main components: a pair of egocentric vision glasses, a central server and a mobile Pioneer P3-DX robot, as shown in Fig. 4.

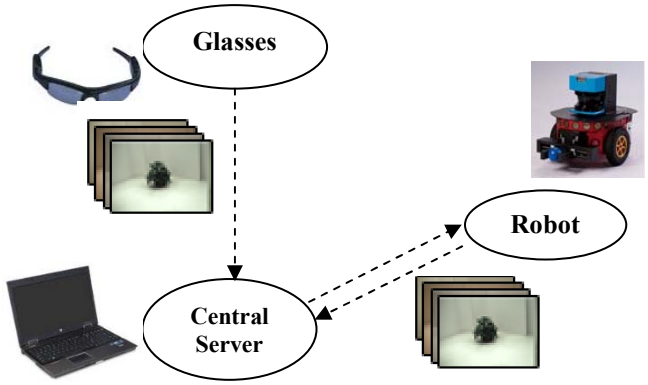

Fig. 4: The main components of the overall assistant co-robot system.

The three components are all connected via a wireless network. The egocentric vision glasses transmit images to the central server (CS) that works as the base of image analysis and commands transmission. The robot shares the camera view as well as the feedback of motions of the egocentric glasses through the CS, which carries the object recognition and motion analysis tasks for the egocentric vision glasses.

A finite-state machine of the system is shown in Fig. 5. Generally, our system starts from an idle mode after power-on. In the idle mode, when the user identifies the target object that he/she wants the co-robot to approach to, the user will turn his/her head around as a circle to trigger the state transition, where the user head movements can be captured by the egocentric vision glasses and analyzed by the CS. Once the turn-around head movement is identified by the CS, the system enters into to the glasses object recognition mode, where the CS begins to recognize the object from images captured by the egocentric vision glasses. When the object has been recognized, the robot

search mode is triggered, which means the robot starts to search for the target object independently using its onboard pan-tilt-zoom camera system.

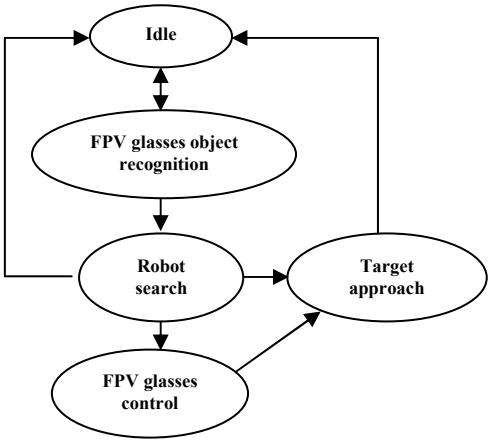

Fig. 5: System finite state machine

If the robot finds the target object, the robot's heading is calibrated, the system switches to target approach mode, and the robot moves towards the object using a path planning technique which tries to find the shortest path to the target object from the current position while avoiding any obstacles on the way. Otherwise, if the robot cannot find the object using its onboard camera within a predefined time period, it sends a "ask for help" signal to the central server via a voice-to-text technique, then the central server switches to the glasses control mode automatically. In the glasses control mode, the robot follows the user's instructions through the user's head movements via the egocentric vision glasses. More specifically, for example, the robot turns left if the egocentric glasses turn left (i.e., the user turns his/her head left wearing the FPV glasses), and does not stop until the egocentric vision glasses stop. After the user guides the robot to the appropriate direction via the movements of the FPV glasses, the user will nod for confirmation, and then the robot jumps to the target approach mode. In the target approach mode, the robot conducts object recognition on its own onboard computer with its onboard camera system.

In the robot search mode and target approach mode, we have a unique head motion pattern for stop any robot's actions: shaking head. If the shaking of egocentric vision glasses is detected by the central system, the robot will stop its current actions and go to a predefined base position staying at the idle mode.

Microsoft Speech SDK 5.1 is employed for the voice-to-text in this system, providing a voice feedback from the robot to the user.

IV. System Components

A. The Egocentric Vision Glasses

The egocentric vision glasses are a pair of sunglasses with a forward looking camera in the middle of it, with a view angle of 60 degrees. It also has a WIFI module to communicate with the CS via 2.4 GHz wireless channel, which is responsible for the transmission of images captured by itself to the CS. We fetch the images in the resolution of 640*480.

The user wearing the egocentric vision glasses looks around, making the view of the glasses camera be roughly equal to the field-of-view of the user. Our system understands the intentions of the user according to these images. In addition, the glasses are equipped with a Li battery which can last for three hours.

B. Motion-pattern Estimation

The Motion-Pattern estimation module uses a 56-dimension motion histogram to estimate the motion pattern of the egocentric vision glasses. The 56-dimension histogram is constructed by one 8-dimensional histogram, one 16-dimension histogram and one 32-dimension histogram to encode the orientation statistics of the flow motion field captured from the egocentric vision glasses. All three histograms are derived from the flow motion fields computed a fixed number of image frames. For example, Fig. 6 shows the histogram of "turn around" motion pattern. The histogram is generated by calculating the moving directions of the glasses based on every two adjacent images captured by the glasses [6].

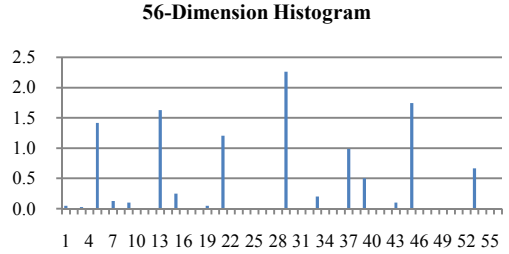

Fig. 6: The histogram of motion pattern "turn around". The first 8 bins show the movement in all eight directions. The 16-dimension histogram, 9th-24th, divides the motion sequence into two partitions: the first records the first half motion's movement directions and the second records the rest. The 32-dimension histogram divides the motion into four partitions.

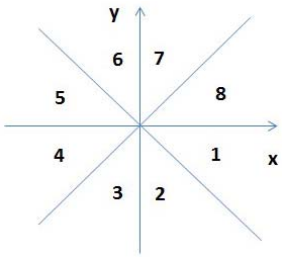

Fig. 7: The division of directions. The first direction is defined as the area within 45 degrees from x-axis in clockwise, the second one within 45 degrees to 90 degrees, and so on.

According to [6], images captured by the glasses are transformed into gray-scale images and then down sampled for a real-time implementation. The Full-search Block Matching

algorithm is applied to these down-sampled images for getting the shifting distance between every two adjacent down-sampled images. A shifting distance is expressed in two dimensions. Particularly, *x* denotes the horizontal shifting distance and *y* denotes the vertical.

Shifting distances are stored in a circular buffer sequentially, from which all histograms are generated. Firstly, all motion vectors in the circular buffer are categorized into eight directions as shown in Fig. 7. A motion vector *(x, y),* is categorized into a specific direction according to the orientation of the vector.

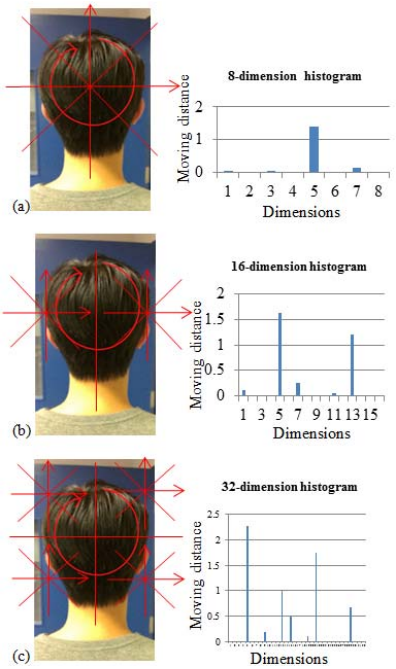

Fig. 8: The extraction of histograms of 8-dimension, 16-dimension and 32-dimension. (a) The 8-dimension histogram is simply generated by categorizing the movement directions and normalizing them. (b)The 16-dimension histogram is simply combined by two 8-dimension histograms. (c) The 32-dimension histogram is simply combined by four 8-dimension histograms.

After the categorization, an 8-dimension histogram has been built: each bin represents the normalization of cumulated sum of vectors' mod in one direction. However, the 8-dimension histogram lost the sequential information, which means that a motion, for example, turn-left after turn-right, will generate the same histogram as a motion turn-right after turn-left. In order to distinguish these motions, more details are required. Therefore, the circular buffer is evenly divided into two parts, which makes it possible for the histograms to keep sequential information. Both two divided circular buffers would generate an 8-dimension histogram similar to the first 8-dimension histogram. The former one represents the first half motion and the latter one represents the rest of the motion. Then a 16-dimension histogram is generated to simply concatenate them together. Finally the circular buffer is divided into four parts and a 32-histogram is generated. By combining all the three histograms we can get a 56-dimension histogram which contains both global and local information. Fig. 8 shows the generation of each histogram. Therefore, each motion can be described by a 56-dimension histogram.

Support vector machine (SVM) are trained on top of the 56-dimension vectors by leveraging a set of labeled training data to recognize different motion gestures. In order to make pattern recognition more robust, some constraints are established. For example, shaking head, of which the y component of the motion vector should be small, is double checked by examining whether the vertical shifting distance is out of a pre-defined threshold or not.

C. Object Recognition

The accuracy of the whole system highly depends on the success of vision based object recognition. We employ the vocabulary tree (VT) based recognition algorithm proposed by Nister and Stewenius [7] for fast indexing and retrieval of relevant object images, and a Homography-based post-verification is conducted to improve the recognition accuracy. 5 images of each object from different views are captured as training samples for the system learning. The SIFT [12] algorithm is applied to extract a set of keypoints and descriptors for each image for VT training. All the descriptors of the objects are considered in a common space in the unsupervised learning of the VT using hierarchical k-means. Hierarchical TF-IDF (term frequency-inverse document frequency) scorings are also established in the database for determining the likelihood of the object appearance in a query image.

The vocabulary tree based indexing discarded the spatial information of the features for efficiency to gain on speed, which increases the chances of retrieving false positive images. Therefore, we use a Homography-based post verification scheme to re-evaluate top 4 candidates from the VT results to make the final decision.

In the re-evaluation, each candidate of the top 4 of the VT results would query the glasses-captured image reversely. These 4 processes are simultaneously executed by multi-threading using OpenMP (Open Multiprocessing), which significantly reduces the execution time by a factor of at least 4, according to the number of threads that can be supported by the CPU. For each computation, SIFT keypoints and corresponding descriptors are extracted at first. The keypoints from two images A and B are matched by the k-nearest neighbor algorithm (k-NN) [13]. All the first best matches are selected as good matches, which are used as the input of Homography transform matrix H. Since matches are always noisy (in other words, wrong matches always are contained in the matching set), RANSAC [14] is applied here to refine the quality from the raw matching set. The transform matrix H is defined in Eqn. (1).

$$p_a = \begin{bmatrix} x_a \\ y_a \\ 1 \end{bmatrix}, p_b' = \begin{bmatrix} w'x_b \\ w'y_b \\ w' \end{bmatrix}, H_{ab} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{31} & h_{33} \end{bmatrix} \quad (1)$$

In order to verify the accuracy of the Homography, several constraints are checked. Frist, we calculate the inverse matrix of *H*, which contains the important information -- scale and angle of a transformation. Typically, the appropriate transformation defines that the scale should be no more than five. In addition, from experimental results, if it fails to generate *matrix H*, the local distribution of *Set A*, which is the set of points wrapped from the object image in the database to the captured image using *matrix H*, would be out of order: either clustering into several dots or spilling over the whole image. Hence, the distribution of object keypoints is checked. The checking windows, defined as the area where *Set A* mostly concentrates, was used to confirm if the amount of *Set A* in this windows is roughly the same as the original keypoints (*Set B*) detected in this area of the captured *Image B*. If the two distributions are matched, norm-*L2* is used to estimate the distance. If keypoints from *Set A* has a corresponding matching point from *Set B* at similar location, the final match can be gained and the ratio of the matching amount over the number of keypoints of *Set B* can be defined as the score of Homography matching in Eqn. (2)

$$S_H = \frac{n_{matched}}{N_w} \quad (2)$$

The score S_H is added to the VT score considered as a final score, if S_H is over 0.7. The final score has been raised by the Homography transform and the object in the *image B* can be detected.

D. Robot Localization, Mapping and Path-Planning

To build up the proof-of-concept assistive co-robot system, we use a Pioneer 3-DX mobile robot as our platform. Pioneer 3-DX is equipped with two rings of sonars (front and back), one SICK laser range finder, one color PTZ camera, and one wireless Ethernet communication card. The laser range finder is used for map building, the sonars are applied for obstacle detection, the color camera is used for object recognition, and wireless Ethernet is used for communication with the central server.

The robot needs to move around in an environment to help the human user. To this end, motion planning is critical for the success of the task. Basically, the motion planning in this robotic system is to control the robot to patrol an area while no target object is found or approach the object when the target is captured and recognized. It generally includes robot localization, map building, and path planning. In this paper, our main goal is not to develop a novel approach to any of these tasks. Instead, we aim to integrate the available techniques in these areas for our assistive co-robot system.

The method first proposed by [15] is applied for simultaneous localization and map building (SLAM). Fig. 9 shows one snapshot of the map built by this SLAM algorithm in our lab environment. Each time when the robot starts up, it boots the laser rangefinder to scan the environment, looking for landmarks [16] that match the map record, grabbing the distances from the landmarks and calculating its coordinates through the triangulation. The motor encoder can help to estimate the current position.

Fig. 9: SLAM Scanning Map.

Once the map is built up, the robot starts the navigation process. In the navigation process, the path planning algorithm will provide an efficient route from the starting point to the destination while avoiding obstacles on the way. Before we apply a path planning method, the scanned map needs to be transformed to a cell-based map representation.

The most popular D* algorithm [17] is applied here for path planning, which aims to search for the shortest path from the current position to the goal. The Process-State function in D* algorithm estimates a shortest path to the goal. The Modify-Cost function updates the cost of nodes which are on the OPEN list. The states are all set to new. The Process-State function is iteratively called until the robot state moved from the OPEN list to the CLOSED list, or the current route is unavailable. The Process-State function points back through the list to the root when the goal is reached or an obstacle is detected. Modify-Cost function updates the cost of nodes on the OPEN list based on the robot's current states.

Figure 10 illustrates how to calculate the destination for the path planning. If the target object has not been found by the robot, the robot moves pan/tilt camera looking around. Once the target object is recognized by the onboard camera or the glasses determine the right direction for the robot in the glasses control mode, the robot executes the path planning algorithm. Finding location of the target can be classified as a mathematical model. The current location of the robot in the 2-demenstional coordination, angleθ, and the distance *d* are easily read from the robot. Based on these three known data, we can always determine

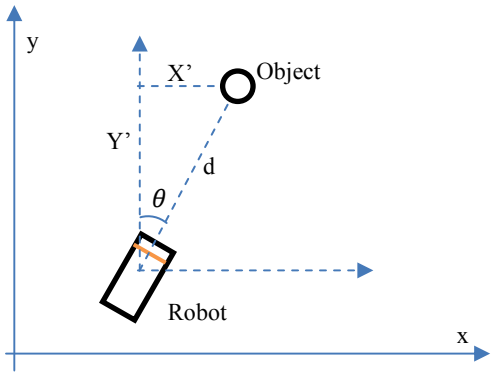

Fig. 10: Finding the Location of Target.

the location of the object geometrically by solving *X'* and *Y'*.

V. EXPERIMENTS

A. Object recognition

We built the unique databases for both the FPV glasses and the robot to compensate for camera differences. Four objects are selected in the prototype system, and for each object we took pictures in five different views as shown in Fig. 11.

B. Motion-Pattern Estimation

Motion patterns used in the project include Nod, Shake and Turn-Around. Histograms of these motion patterns are collected and trained with a multi-class SVM. There are 180 positive samples and 322 negative samples. Each motion pattern (Nod, Shake or Turn-Around) has 60 positive samples. Both positive and negative samples are divided into a training set and a testing set. The training set consists of 40 positive samples for each pattern and 266 negative samples, and the rest of the samples are used for the testing set. The average prediction is 93.650% for all patterns, the accuracy in recognizing each pattern is in Table 1.

TABLE 1: ACCURACY OF RECOGNIZING EACH MOTION PATTERN

	Nod	Shake	Turn-Around
Accuracy	80%	95%	85%

The experimental results show that the motion recognition accuracy is sufficiently good.

Fig. 11: Training Images database, the first 4 rows are the trainning images from the eogocentric vision glasses, and others are the training images from the robot.

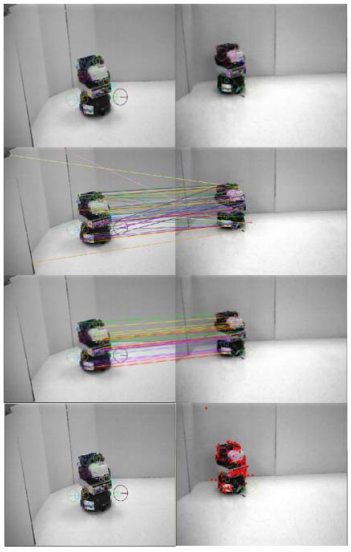

Fig. 12: The first column is the retrieved image from the database, and the second column is the image captured from the camera of the egocentric vision glasses. The first row shows that SIFT keypoints have been extracted. The second row shows k best matches have been identifed. The third row indicates that RANSAC is used to refine the matching result and matrix H is generated. The last row shows the projection is performed and final confirm has been made.

C. System Evaluation

The system is implemented in a square home-like environment containing the robot, target objects and obstacles.

Fig. 13: Experimental Area

Three searching strategies are applied for the system evaluation: Random Search (RS), where the robot moves toward a random direction if it cannot find the target object, Tour Search (TS), where the robot tours around several predefined locations of the environment to find the target object, and the user-guided approach presented in this paper, where the user guides the robot to the correct direction when needed and makes the robot search the target on the direction. Note both RS and TS are running in a fully autonomous mode without engaging the user in the loop. In order to objectively evaluate the performances of these three different approaches, the robot starts at a fixed home point, while

positions of objects and obstacle are different for each experimental run. During each experimental run, the experimental setting is exactly the same for three different approaches for comparison. An obstacle is placed between the robot and target object in the experiments in the second set of experiments, listing in the second row of Table 2.

TABLE 2: EXPERIMENT RESULTS (UNIT: SECOND)

Finding& moving to target object	RS	TS	our approach
without obstacle	173.221	67.927	29.923
with obstacles	252.172	124.222	64.083

300 seconds are set to be the time-out upper bound for all the object finding experiments. In other words, if the robot cannot find the target object within 300 seconds, the robot will ask for help. The average time consumptions for each strategy under different scenarios are listed in Table 2. The results came from the average of 30 different experimental runs. Our user-guided approach obviously excels the other two. It's 3 times and 6 times better than tour search and random search, depending on whether there are obstacles presented or not.

VI. CONCLUSION

In this paper, we present a proof-of-concept egocentric vision based assistive co-robot system, which actively engages the user in the loop for fulfilling tasks. Our experiments in an object finding task clearly demonstrated the efficacy of the human-in-the-loop strategy. Our future work will investigate more principled uncertainty based active learning scheme to engage the users in the loop, beyond the time-out criterion.

REFERENCES

[1] G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Magaritis, M. Montemerlo, J. Pineau, N. Roy, J. Schulte, and S. Thrun, "Towards Personal Service Robots for the Elderly," in *Workshop on Interactive Robots and Entertainment (WIRE 2000)*, 2000

[2] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, "Towards robotic assistants in nursing homes: Challenges and results," Special issue on Socially Interactive Robots, Robotics and Autonomous Systems, vol. 42, no. 3 - 4, pp. 271 - 281, 2003.

[3] P. Hoppenot, E. Colle, O. A. Aider and Y. Rybarczyk, "ARPH-assistant robot for handicapped people-a pluridisciplinary project," in International Workshop on Robot and Human Interactive Communication, Bordeaux, Paris, 2001.

[4] M.Topping and J.Smith, "The development of Handy 1, a robotic system to assist the severely disabled,," Rehabilitation Robotics, p. 244, 1999.

[5] B. Graf, M. Hans and R. D. Schraft, "Care-O-bot II—Development of a Next Generation Robotic Home Assistant," Autonomous Robots, vol. 16, no. 2, pp. 193 - 205, 2004.

[6] J. Wang, S. Zhai and J. Canny, "Camera phone based motion sensing: interaction techniques, applications and performance study," in UIST '06 Proceedings of the 19th annual ACM symposium on User interface software and technology, 101 - 110, 2006.

[7] D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference, 2006.

[8] Y. Takahashi, N. Hasegawa, K. Takahashi and T. Hatakeyama, "Human interface using PC display with head pointing device for eating assist robot and emotional evaluation by GSR sensor," in Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, 2001.

[9] K. Kawamura and M. Iskarous, "Trends in service robots for the disabled and the elderly," in Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems 'Advanced Robotic Systems and the Real World', IROS '94. , 1994.

[10] S. Y. Choi, C. D. Anderson, J. D. Glass and C. C. Kemp, "Laser pointers and a touch screen: intuitive interfaces for autonomous mobile manipulation for the motor impaired," in Assets '08 Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility, 2008.

[11] H. Nguyen, A. Jain, C. D. Anderson and C. C. Kemp, "A clickable world: behavior selection through pointing and context for mobile manipulation," in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008.

[12] D. Lowe, "Object recognition from local scale-invariant features," in the Seventh IEEE International Conference, 1999.

[13] D. Bremner, E. Demaine, J. Erickson, J. Iacono, S. Langeman, P. Morin and G. Toussaint, "Output-Sensitive Algorithms for Computing Nearest-Neighbour Decision Boundaries," Discrete & Computational Geometry, vol. 33, no. 4, pp. 593 - 604, 2005.

[14] F. A. Martin and B. C. Robert, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM, vol. 24, pp. 381-395, 1981.

[15] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," in IEEE Transactions on Robotics and Automation, 1991.

[16] S. Se, D. Lowe and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. The international," International Journal of Robotics Research, vol. 21, pp. 735 - 758, 2002.

[17] A. Stentz, "Optimal and efficient path planning for partially-known environments," in IEEE International Conference on Robotics and Automation, 1994.