Mobile Manipulation - A Challenge in Integration

Cressel Anderson, Ben Axelrod, J. Philip Case, Jaeil Choi, Martin Engel, Gaurav Gupta, Florian Hecht, John Hutchison, Niyant Krishnamurthi, Jinhan Lee, Hai Dai Nguyen, Richard Roberts, John G. Rogers, Alexander J. B. Trevor, Henrik I. Christensen, Charles Kemp

Robotics and Intelligent Machines @ GT
Georgia Institute of Technology
85 5th Street, Atlanta, GA 30332-0760

ABSTRACT

Mobile manipulation in many respects represents the next generation of robot applications. An important part of design of such systems is the integration of techniques for navigation, recognition, control, and planning to achieve a robust solution. To study this problem three different approaches to mobile manipulation have been designed and implemented. A prototypical application that requires navigation and manipulation has been chosen as a target for the systems. In this paper we present the basic design of the three systems and draw some general lessons on design and implementation.

1. INTRODUCTION

Robotics has made tremendous progress in the last four decades in terms of design of manipulation systems and deployment of mobile robots. However, limited progress has been reported in mobile manipulation, primarily due to lack of adequate platforms. The required combination of power, weight and real-time control has not been available until recently. Today it is possible to access light weight arms, and there are a number of mobile platforms that are small and powerful enough to use with a manipulation system. Use of such systems allows for a variety of new applications both in terms of manufacturing as well as service robotics to assist people in everyday situations. A challenge in the design of such systems is tight integration to achieve real-time performance. There are quite a few methods available for construction of systems but few have considered the integration for real applications. To study this problem, we selected a scenario where the system has to deliver coffee to a person. As part of the scenario, the system has to locate the cup, pick up the cup, navigate to the coffee maker, operate the coffee maker and finally deliver the coffee to the human. The key competencies required for this task are also applicable to a large variety of everyday tasks such as preparation of meals, fetch-and-carry tasks, and basic assistance. The platform consisted of a lightweight KUKA KR5 sixx R650 manipulator mounted on a Segway RMP 200 mobile base. The dynamically stable Segway base was chosen to make the task more interesting. In this configuration, the movement of the arm influences the motion of the platform, so there is a need to carefully consider integration of the two systems. In many real-world applications this might not be a realistic platform, but it is an ideal setup to truly demonstrate the challenges of mobile manipulation. This particular evaluation was performed in an educational setting where three groups of students each undertook the particular task using different strategies. This paper reports some of the lessons learned and issues to be considered in design of future systems. The paper is organized with a review of the relevant literature in Section 2, followed by a brief presentation of the platform and the hardware architecture with the three specific approaches in Section 3. Section 4 briefly discusses the results obtained with the systems, and Section 5 provides a summary and a discussion of future challenges in Section 6.

2. RELATED WORK

There has been a significant amount of previous work on addressing a variety of mobile manipulation tasks. Some early work on behavior-based mobile manipulation was done using a platform named Herbert,[7] which is a statically stable mobile base equipped with a custom built arm. This robot is capable of searching cluttered human environments for soda cans and grasping them.

Additional work on behavior-based controller architectures for mobile manipulation was done by Petersson *et al.*,[1] which was later expanded upon.[2] Formal methods for sequencing complex tasks and assigning processes to

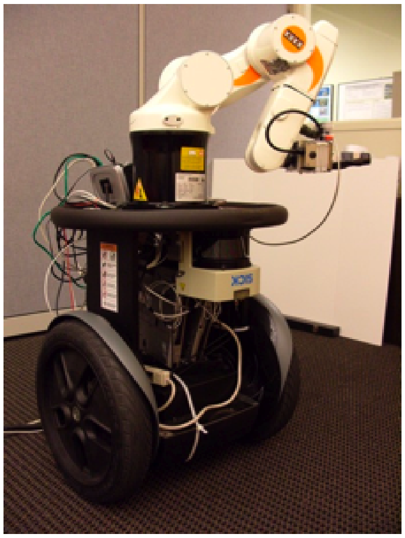

Figure 1. The platform described here, including Segway RMP and KUKA KR5-Sixx.

resources were established in the Distributed Control Architecture[3] for the application of mobile manipulation. Petersson *et al.* have emphasized scalability and availability in a real-time system by using a tree hierarchy with as much local computation and spatial locality as possible.

Another example of a successful mobile manipulation platform is the CARDEA robot developed at MIT.[5] This platform consists of a Segway RMP mobile base, and a custom-made compliant manipulator. The robot is capable of navigating halls, identifying doors, and opening doors. This platform is relevant to the work presented here because it also uses a Segway RMP mobile base.

Mobile manipulation systems have also been implemented with the aim of human-robot cooperation. Romeo and Juliet[4] use real-time control and force sensing to augment human efforts for moving large or heavy objects. These statically stable mobile platforms are equipped with Puma arms and force-torque sensors, allowing them to engage in cooperative mobile manipulation tasks either with each other, or with a human.

The HERMES[6] robot is capable of mobile manipulation grasping tasks. This robot includes a statically stable mobile platform, and two custom built arms. It is capable of grasping several objects, such as a bottle and a glass. Several other examples of service robot platforms for mobile manipulation exist.

3. APPROACHES

As part of this course, students formed three teams. All teams shared the same hardware platform, but chose different techniques and strategies for accomplishing the task. The teams also chose different software platforms and tools. We present an overview of the approaches taken, focusing on the strategies that were most successful in this mobile manipulation task.

3.1 Approach 1

Team 1 utilized Player/Stage[8] to interface with the platform hardware, a particle filter for robust visual pose estimation of the objects, and KUKA Remote Sensor Interface (RSI) to control the arm. These components communicated via TCP sockets, and were controlled by a Java front-end. The system used two Core 2 Duo laptops running Ubuntu Linux. The programmed task was initiated and monitored with an additional two laptops at a station outside of the testing area. Using this approach, the team was able to successfully complete three runs robustly in rapid succession during the final demo.

A Player server ran on one of the two laptops and provided the interface to the Segway and SICK laser scanner. Localization and navigation components were also implemented[9].[10] Because the approximate locations

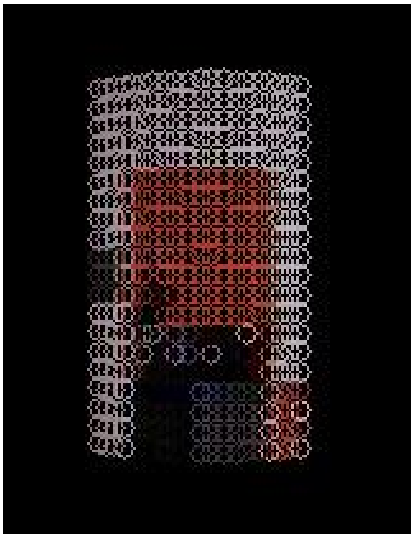

Figure 2. Team 1: Object model of the cup.

of the coffee mug, coffee machine, and coffee delivery location were known, the navigation controller was able to move the platform to the desired pose, and then transfer the control to the vision system and manipulation controller. Both position and velocity commands were used to control the end effector position.

The success of this team's approach was largely due to the robust operation of the vision system. A model based Monte Carlo approach[11] was used in conjunction with the constraint that the cup would be oriented approximately vertically. First, models of the coffee cup and coffee maker were created by sampling approximately 300 points on the 3D geometry of the cup, projected into a training image to yield color values for each three dimensional point on the surface, as seen in Figure 2. In this case, the points were created using physical measurements of the objects dimensions. Next, using an appropriately tuned particle filter, the object's pose could be quickly and accurately be determined, as seen in Figure 3.

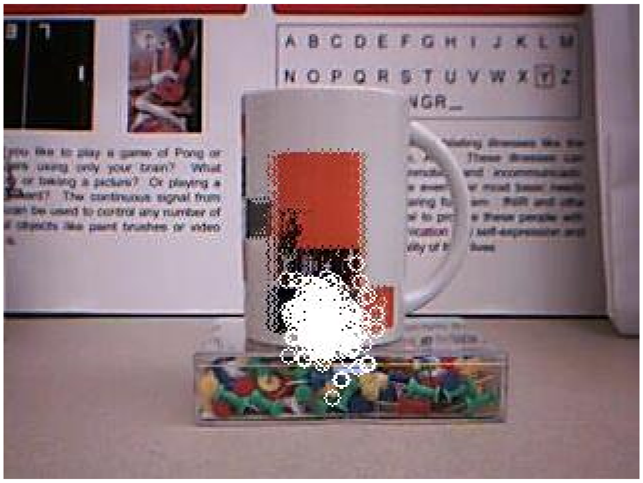

Figure 3. Team 1: Pose estimation of the cup.

The models used in this approach made use of the known appearance of the cup and coffee machine. Because color was used, varying lighting conditions and specular features could interfere with object detection. To counter the specular features on the metal surface of the coffee machine, a feature was added to the model building program to mask off areas prone to specular features. This addition removed the areas of brushed aluminum on the coffee maker face from the model. For the objects used in this task, model construction was simple as the cup is well approximated with a cylinder and the face of the coffee maker is well approximated by a plane. Modeling objects of arbitrary geometry was not addressed and is problematic for this technique.

Because the vision system made use of the camera calibration, it reported object pose in real-world units. This greatly sped up programming of end effector positions with respect to the tracked objects by allowing the team to simply measure distances in the real world, followed by minor tweaking of coordinates.

The dynamic stability of the Segway added complexity to the task of visual servoing. As the arm was extended, the center of gravity of the platform would shift accordingly, and the platform would roll forward or backward to accommodate this shift. To deal with these large unmodeled movements, multiple closed-loop controls were employed, running simultaneously at several levels of body control. One controller moved the platform as its center of balance changed, to attempt to keep the arm coordinate frame stationary in the world. A second controller servoed the arm to be directly in front of the object to be grasped. At very close distances, the vision system's estimate of the object pose was used to continuously servo the end effector to the target pose. This multi-layered controller helped make performance robust even in these dynamic manipulation tasks.

3.2 Approach 2

Team 2 chose to use Microsoft Robotics Studio as their software platform. This choice simplified inter-process communication and simplified modularization of components. The vision module was written in C++, which required integration of an unmanaged library into MSRS. Several hardware drivers needed to be written for this new platform, but once they were written, orchestrating them was a simple matter. MSRS also provided message logging, state inspection, and debugging.

Team 2 implemented a unique localization routine in MSRS. They used a Hough transform to locate straight walls and adjust the robot's distance from them. This allowed the robot to navigate to the approximate locations of the objects of interest.

The vision system tried to detect the cup in an arbitrary 3D pose, allowing for suboptimal relative poses. The repeatedly-estimated 3D poses of the cup are fed back to the visual servoing module, which interpolates the current and target poses of the gripper in SE(3) space. While this approach allows the vision and grasping to be suitable for more general situations, vision system is prone to more deformation of the visual pattern on the cup.

Robust object recognition was achieved by segmenting the image using superpixels combined with edges. Then a representative combination of orange, black, and gray blobs are found. The homography can be easily calculated using the four corners and the physical dimensions of the pattern on the cup. This approach produced more robust results in our particular environment settings than SIFT[12] features, especially when the cup is observed from above. Figure 4 shows an example of the visual detection of the cup.

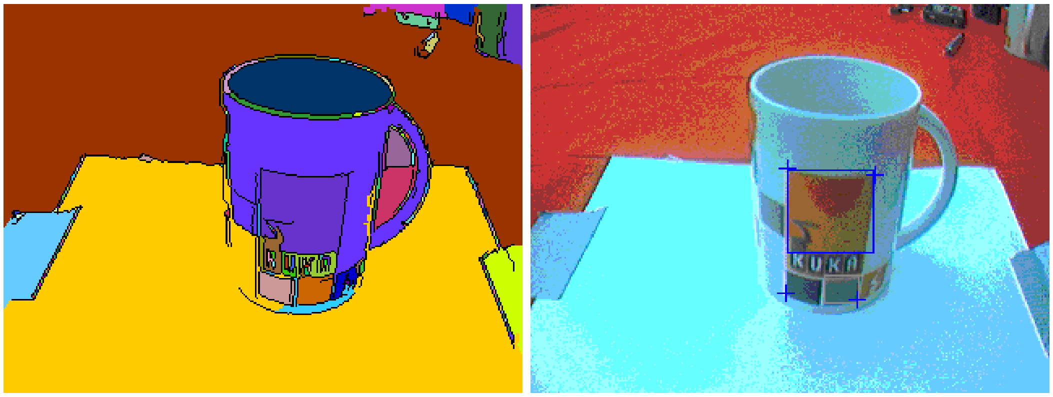

Figure 4. Team 2: Estimation of the 3D pose of the cup.

To improve the visual servoing, this team used the Segway base to position the arm directly in-front of the target before attempting to grasp. This eliminated undesirable motion of the Segway due to lateral arm motion. A proportional controller was also implemented to keep the base of the arm in a fixed location during grasping.

3.3 Approach 3

Team 3 implemented a system using Player/Stage,[8] C++ and OCaml. Player drivers were written for several hardware components for which they did not already exist, including the KUKA arm, Schunk Gripper, and Segway RMP (USB). These drivers were used by a navigation module and a visual servoing and manipulation module in order to control the robot. The AMCL module built-in to Player-Stage[8] was used for localization[10].[9] A high level control module managed the overall task execution.

The highlight of this team's approach was their sensing and manipulation strategy. Object recognition and pose estimation were performed using the SURF feature detector.[13] This detector, which is similar to the SIFT[12] feature detector, was found to provide more robust features and was more efficient. Feature correspondences were computed between a model image of the object of interest and the scene image from the robot's camera. A homography was robustly established in a least squares sense between the model image and the matched scene features using DLT[14] and RANSAC.[15] The homography gives a measurement in image coordinates of the center of the object, using a planar assumption. The planar object assumption worked well for the front face of the coffee machine, but not as well on the coffee cup because it has a cylindrical geometry. The homography also establishes a range measurement since the ratios of distances between features is preserved under perspective projections. This bearing and range measurement was then rotated into the KUKA arm's base coordinate frame where it is used to inform a particle filter[10] . The world coordinate frame was not used because this team assumed that the base would be stable enough during the task to consider each component in a local coordinate frame without unifying them in a globally consistent frame. The SURF feature detector was robust to changes in lighting conditions and could even detect the coffee maker with up to 40 degree rotation out of the plane.

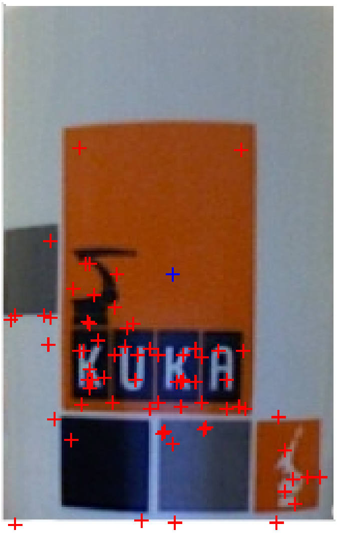

Figure 5. Team 3: SURF Features detected on the coffee cup.

A visual search for the object of interest was performed by panning the camera across the scene. As the Segway RMP is a dynamically stable platform, arm movements altered the balance of the platform, causing it to move to maintain it's balance. If the Segway moved significantly during a grasp attempt, the grasp would fail because these movements were not taken into account. To increase chances of grasp success, this group tried to ensure that the base was well stabilized by not attempting grasps until the particle filter's position estimate converged sufficiently and the base had enough time to stabilize. After a grasp motion was attempted, the visual search was repeated to see if the cup remained in the scene. If the cup is still on the table instead of in the gripper, then the grasp is attempted again.

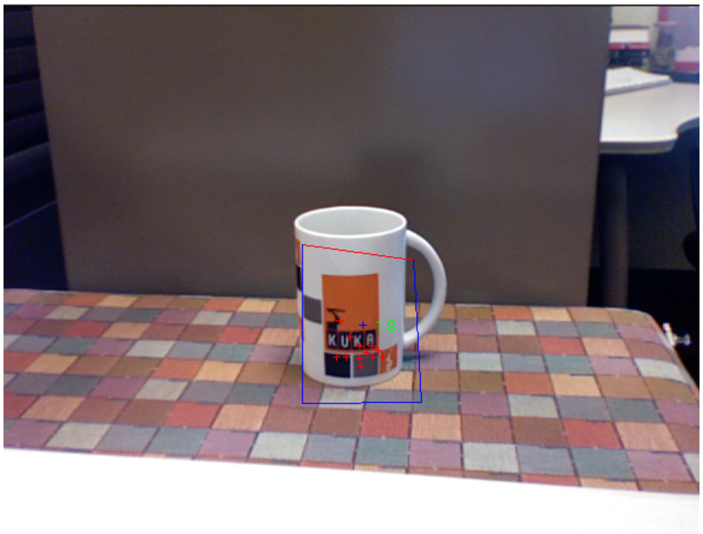

Figure 6. Team 3: Homography matched to the coffee cup

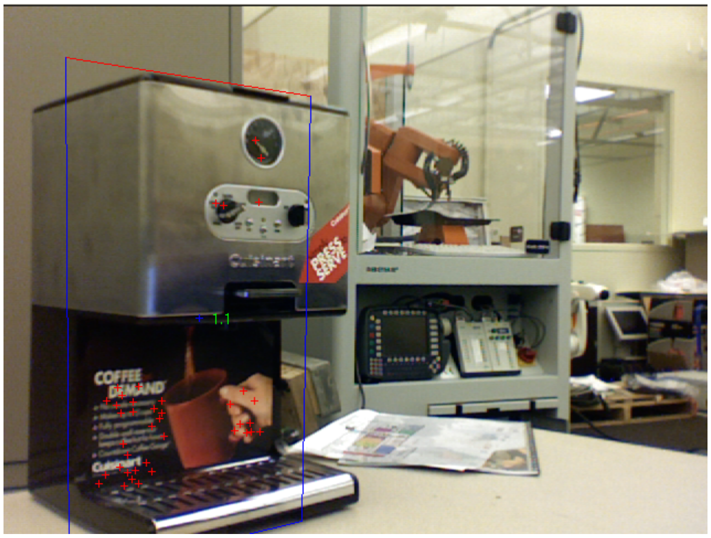

Figure 7. Team 3: Homography matched to the coffee machine.

4. RESULTS

The three implementations of the system were briefly compared by testing how many cups of coffee could be served autonomously in 20 minutes. Ideally, the process would proceed as follows. The system was powered on at a known location. From there, the robot would locate a coffee mug and grasp it. The robot would then drive to the coffee machine, place the mug under the dispenser, and press down on the lever to dispense coffee into the mug. Finally, the mug would be grasped again, and placed on a nearby table at the delivery location. The approximate location of the mug and coffee machine were know *a priori*, but object pose recognition was still required in order to interact with these objects.

Team 1 was able to successfully serve three cups of coffee during a demonstration of the system. The vision system was re-trained prior to the demonstration to ensure that the object models used current lighting conditions. The multi-layered controllers describe previously helped make performance robust, even in this dynamic manipulation task.

Team 2 was able to successfully locate and grab the cup, then bring it to the approximate location of the coffee maker. Due to time constraints, the feature detector for the coffee maker was incomplete, so they were unable to recognize it and dispense coffee.

Team 3's system was able to successfully perform the coffee serving task as specified. The sensing and manipulation strategy the team used was effective, and was able to reliably recognize and manipulate both the cup and the coffee machine if the Segway RMP platform was well positioned. However, the base was sometimes poorly positioned relative to the coffee cup or coffee machine, which would cause the manipulation of the object to fail, so the team's approach would have benefited from servoing the base in addition to the KUKA arm in these cases.

5. DISCUSSION

One of the most salient conclusions we can draw from our experiences here is the need for a tight integration between the control systems of the mobile platform and the manipulator. The Segway is often making adjustments to it's pose in order to remain upright because it is a dynamically stable platform. Moving the arm shifts the center of mass of the robot, requiring the platform to move to compensate for the loss of balance. In all three approaches, the base coordinate system for the arm was considered to be horizontal at all times. In the evaluations it was evident that this was a poor design choice. This introduced a rotation into the pose of the object which was not modeled, therefore the pose of the object was less accurate when no longer observed. Care was taken to minimize the amount of platform movement when servoing the end-effector to grasp the object, but all of the systems would benefit from a tight integration between platform pose (4D) and arm control. None of the implemented approaches used a real-time operating system which would have been necessary to integrate the control of the mobile base and the manipulator.

In terms of navigation the environment was assumed to be well approximated by a two dimensional representation. In such environments navigation is largely a solved problem and solutions are available to provide an accurate localization. The environment was relatively simple and only two objects had to be recognized. Despite this small number of objects of interest, variability in illumination, texture, and background, this is still a challenge. A variety of different feature detectors were tested including some of the most widely used such as SURF, SIFT, and template matching; all of them would fail under mildly non-ideal imaging conditions. Design of robust visual methods is still a challenge. One of the primary differences between the three systems is their approach to object recognition. As described above, Group 1 used a detailed model of the coffee cup consisting of expected color at points on the surface of the cup. This has the advantage of being quite invariant to various relative poses between the camera and the cup due to the use of approximated geometry. The use of color features resulted in poor tolerance to lighting variation and a need to retrain the model often, but was sufficient for the task.

Instead of using SIFT or SURF features, Group 2 uses a super-pixel approach to find the color blobs on the KUKA cup. A bounding box is placed around these, and the corners of the bounding boxes of these regions are used as features. A homography between these corner features and the model image is calculated, yielding an estimate of the pose of the cup.

Group 3's approach assumes the object of interest is planar, and attempts to find a homography between the features detected in the scene and the model image. Even though the coffee cup is not actually planar, this approach provided a reasonable approximation of the coffee cup's location which was accurate enough to grasp the coffee cup. An advantage of this approach was the ease of generating a model. All that is required to generate a model with this system is to take an image of the object of interest, and measure the distance to the image of interest in the model image.

6. OPEN CHALLENGES

The presented study is a good example of the type of systems that are about to emerge. Mobile manipulation has been hampered by access to adequate hardware. With the introduction of light weight manipulators such as the Barrett arm and the KUKA Light weight arm (LBR) it is feasible to deploy manipulators on mobile platforms. Integration of mobility and manipulation at the same time offers a much richer flexibility than seen earlier and as such opens new application areas. However, this flexibility comes at a cost. There is clearly a need to perform integrated control of the two components to achieve efficient motion control.

Over the last 5 years significant progress has been made in visual recognition which has enabled the use of techniques beyond simple 2D shape matching. At the same time, increases in CPU power have allowed implementation of these techniques in real-time. Visual servoing has opened up flexibility in terms of control of the robot based on the position of objects in the environment.

The challenges that remain are generalizing these task-specific methods to more open-ended environments. This paper presents examples of systems that operate in environments with a simple wall layouts and recognize only 2 objects (cup and coffee maker). To generalize from a set of specific examples to an entire object class is an open research problem.

In addition, it was assumed that the environment was static, which is a strong assumption. Also, none of the reported systems used a formal development process. Today there is a gradual availability of software tools for development that allow specification and realization while also allowing for verification. Safety is essential in any robot application and as such the use of such formal processes and tools will be essential for management of complex systems.

Another challenge in mobile manipulation is integration. Many mobile manipulation platforms, such as the one presented here, involve augmenting a mobile platform with a manipulator. The mobile platform and the manipulator each have their own control system, and integrating these to produce the desired outcome can be challenging. For some applications, it may be sufficient to move the mobile platform to a desired location, and then perform some manipulation. In this case, the mobility portion and the manipulation portion of the task can be treated separately. However, some tasks will require a closer coupling of these control systems.

ACKNOWLEDGMENTS

The authors would like to acknowledge KUKA Robotics for supplying the manipulators used in this work. Team 3 would also like to thank Herbert Bay for providing Mac binaries of his SURF feature detector.

REFERENCES

[1] L. Petersson, M. Egerstedt, and H. Christiensen, "A hybrid control architecture for mobile manipulation," *Intelligent Robots and Systems* , 1999.

[2] H. Christensen, L. Petersson, and M. Eriksson, "Mobile manipulation: Getting a grip?," *International Symposium of Robotics Research* , 1999.

[3] L. Petersson, D. Austin, and H. Christiensen, "DCA: A distributed control architecture for robotics," *Intelligent Robots and Systems* , 2001.

[4] K. Chang, R. Holmberg, and O. Khatib, "The augmented object model: Cooperative manipulation and parallel mechanism dynamics," *International Conference on Robotics and Automation* , 2000.

[5] R. Brooks, L. Aryananda, A. Edsinger, P. Fitzpatrick, C. Kemp, U. O'Reilly, E. Torres-Jara, P. Varshavskaya, and J. Weber, "Sensing and manipulating built-for-human environments," *International Journal of Humanoid Robotics* , 2004.

[6] R. Bischoff and V. Graefe, "Hermes - a versatile personal assistant robot," *Proceedings of the IEEE - Special Issue on Human Interactive Robots for Psychological Enrichment* , 2004.

[7] R. Brooks, "Elephants don't play chess," *Robotics and Autonomous Systems* , 1990.

[8] B. Gerkey, R. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," *Proceedings of the 11th International Conference on Advanced Robotics* , 2003.

[9] K. Konolige, "Markov localization using correlation," *Proceedings of the International Joint Conference on Artificial Intelligence* , 1998.

[10] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Efficient position estimation for mobile robots," *National Conference on Artificial Intelligence* , 1999.

[11] R. Douc, O. Cappe, and E. Moulines, "Comparison of resampling schemes for particle filtering," *Image and Signal Processing and Analysis* , 2005.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision* , 2004.

[13] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Proceedings of 9th European Conference on Computer Vision* , 2006.

[14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.

[15] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM* , 1981.