A New Mobile Manipulation Platform for Automatic Coffee Retrieval

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

***Abstract*—A new platform for mobile manipulation consisting of a Segway RMP base and a KUKA KR-5 sixx manipulator was developed at the Georgia Institute of Technology in the context of a class. Students formed three teams whose goal was to design a system capable of autonomously serving coffee. Each team took a different approach to the problem in terms of system architecture, visual recognition, and grasping procedure. The approaches used by the students and their merits with respect to this task are presented.**

I. Introduction

This platform was developed as part of a course titled "Mobile Manipulation" offered in Fall 2007 at the Georgia Institute of Technology, which was taught by Henrik I. Christensen and Charlie Kemp. The platform used in this class consisted of KUKA KR-5 sixx R650 industrial robot arm mounted on a Segway RMP200 base. The end effector was a Schunk PG-70 2-finger parallel gripper. A Unibrain Fire-i firewire camera was mounted on the end effector for use in visual servoing. The platform was also equipped with a SICK LMS291 for localization and navigation. The platform is shown in figure 1.

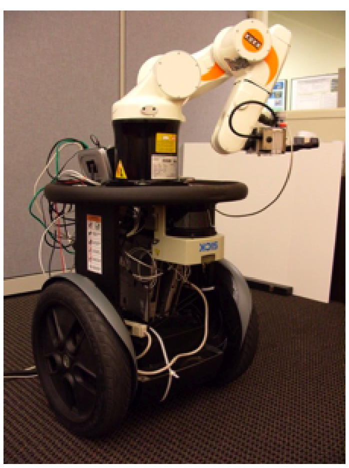

Fig. 1. The new mobile manipulation platform

This platform was developed for the task of autonomously serving coffee. Specifically, the robot's task was to find a cup, take it to a coffee machine, fill it with coffee, and then return the mug to a delivery location. The appearance of the cup and the coffee machine were known. The approximate location of the cup, coffee machine, and delivery location were also known.

II. 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.

A. Team 1

Team 1 utilized Player/Stage [2] 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.

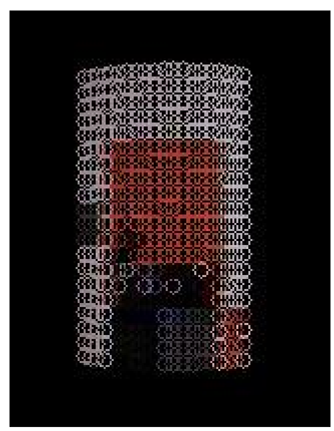

Fig. 2. Team 1: Object model of the cup.

A player server ran on one of the two laptops and provided the interface to the Segway and SICK laser. Localization and navigation components were also implemented. Because the approximate locations of the coffee mug, coffee machine, and coffee delivery location were known, the navigation controller was able to have the platform achieve the desired pose, and then transfer the control to the visual manipulation controller for grasping and manipulation. Both position and velocity commands were used to control the end effector position. The success of the team's approach is largely due to the robust operation of the vision system. A model based Monte Carlo approach was used in conjunction with the constraint that the cup would be oriented vertically. First, models of the coffee cup and coffee maker were created by sampling an image of the object and transforming the image to yield 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 objects pose could be quickly and accurately be determined, as seen in Figure 3. Additionally, the arm configuration was utilized in the particle filter for updating the hypotheses to account for changes in the camera pose, thereby enabling better dynamic tracking of objects.

Fig. 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 mask creation feature was added to the model building program. This addition removed the areas of brushed aluminum on the coffee maker face from the model.

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.

B. *Team 2*

Team 2 chose to use the new Microsoft Robotics Studio as their software platform. This choice simplified inter-process communication and made for easy modularization of components. Although not required, they also ported their computer vision code to work inside of MSRS as unmanaged C++ code. 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 easy message logging, state inspection, and debugging in Visual Studio.

Because MSRS does not include a SLAM package, team 2 had to implement an ad-hoc localization routine. They used a Hough transform to locate straight walls and adjust the robot's distance from them. This significantly helped the visual servoing.

Robust object recognition was achieved by segmenting the image using superpixels combined with edges. Then a representative combination of orange, black, and grey blobs are found. The homography can be easily calculated using the four corners and the physical dimension of the pattern on 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.

C. *Team 3*

Team 3 implemented a system using Player/Stage [2], 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. A third module managed overall task execution and planning. These drivers were used by a high level control module which managed the overall task execution.

The highlight of this team's approach was their sensing and manipulation strategy. Object recognition and pose estimation was performed using the SURF feature detector[1], which is similar to the SIFT feature detector but was found to more reliably match features for this task. Feature correspondences were computed between a model image of the object of interest and the scene image from the robot's camera. A model image with minimal specular features was used because these are dependent on lighting. A planar homography between the detected features and the model image is then computed using DLT and RANSAC. The planarity assumption worked well for the front face of the coffee machine, as well as the cylindrical coffee cup, even though the cup was not actually a planar surface. A particle filter was then used to track the pose of the object of interest in the KUKA arm's coordinate frame. The world coordinate frame was not used due to difficulties in incorporating the tilt of the Segway platform, though it is likely that this would have improved performance. This approach was robust enough to work with changes to the lighting conditions and orientation of the object. For example, the coffee machine could be recognized robustly even under a 40 degree rotation.

Before attempting to grasp the cup or operate the coffee machine, the robot servoed the arm in a plane parallel to the camera's image plane such that the object of interest was in the desired position relative to the arm. As the Segway RMP is a dynamically stable platform, arm servoing induced motion in the platform that could cause graps to fail. To increase chances of grasp success, this approach tried to ensure that the base was well stabilized by not attempting grasps until the particle filter's position estimate converged sufficiently. Grasp failures were detected after an attempted grasp by surveying the scene for presence of the object that the robot attempted to grasp. It is assumed that if the object is present then a grasp failure has occurred, triggering another attempt at the grasp.

III. Results

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 benefitted from servoing the base in addition to the KUKA arm in these cases.

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc J. Van Gool. Surf: Speeded up robust features. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *ECCV (1)*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer, 2006.

[2] Brian Gerkey et al. *The Player Robot Device Interface*, 2006. Online document from http://playerstage.sourceforge.net/doc/Player-2.0.0/player/.