Toward Autonomous Sampling and Servicing with the Ranger Dexterous Manipulator

Mike Naylor[*], Nick Scott[*], Ella Atkins[†] and Stephen Roderick[‡]

Space Systems Laboratory, University of Maryland, College Park, MD, 20742, USA

Future space and planetary surface robotic missions will require increased autonomy levels to manage substantial time delays and improve long-term operational efficiency. To enable autonomous manipulation, robotic systems must accurately identify grasping targets and must plan and robustly execute trajectories between them. This paper describes an extension of the University of Maryland's Ranger manipulator system to autonomously identify, locate, and retrieve known objects suspended within the manipulator's workspace. A machine vision system identifies target locations with a stereo camera pair rigidly affixed to the Ranger body, and then a trajectory planner generates smooth joint-space motions to the target while avoiding workspace singularities. Experimental results demonstrate Ranger's ability to reliably retrieve the target with a variety of positions and lighting conditions. After the target has been grasped, a final scripted action sequence is executed to stow the manipulator.

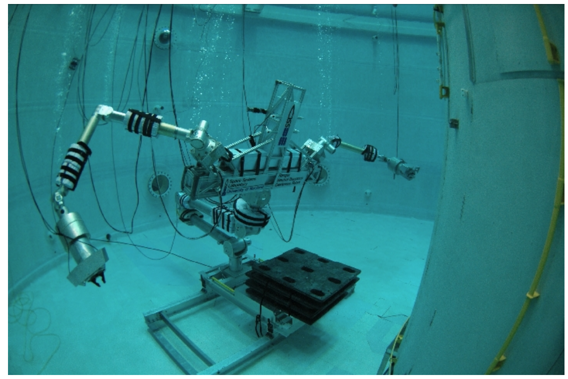
I. Introduction

Robotic systems are expected to greatly enhance our ability to explore the Moon, Mars, and beyond. However, all such systems, launched to-date, have relied on teleoperation as their primary means of control, requiring extensive ground support and high operator workload, especially with time-delayed feedback data. Preparation of the Moon and Mars for human presence will require numerous robotic explorers to be deployed in space and on planetary surfaces. Mission operations overhead will be prohibitive, however, unless each robot only requires infrequent supervision rather than constant control. Technologies are under development to support autonomous operations. However, the evolution toward increased autonomy levels must enhance exploration capabilities without increasing risk. Extensive testing in a variety of Earth-based environments is critical before propelling such systems into the unforgiving space environment.

The University of Maryland Space Systems Lab (SSL) has developed a variety of free-flying and manipulation-based telerobotic systems and tested them in neutral buoyancy to simulate the space environment. The Ranger telerobotic manipulation system[1] operates in either 1-G or neutral buoyancy environments. Ranger is designed for on-orbit servicing of spacecraft and satellites, a task requiring multiple manipulators to grapple a satellite or component, provide video feedback to operators, and provide tool-based operations on components being serviced. The robot consists of a central body, which houses the main computers and electronics, and serves as a base platform for all the manipulators, any subset of which can be used during a test. Ranger has two eight degree-of-freedom (DOF) dexterous manipulators for object manipulation and a 7-DOF video manipulator. It also has a 6-DOF positioning leg that anchors it to a fixed base, which could be modified to enable grappling or docking to a spacecraft. Shown in Figure 1, Ranger supports two different mechanical configurations. In its normal configuration, the manipulators mount directly to a small electronics housing. The dexterous manipulators and positioning leg are 1.3 meters and 2.0 meters in length respectively as in Figure 1(a). In its wide-body configuration, Figure 1(b), the central body incorporates a large T-shaped frame with each manipulator mounted approximately 1 meter apart. Each dexterous manipulator is approximately 2.7 meters long, while the positioning leg is approximately 5.0 meters long.

[*]Graduate Research Assistant, Department of Aerospace Engineering, Student Member.
[†]Assistant Professor, Department of Aerospace Engineering, Senior Member.
[‡]Faculty Research Assistant, Department of Aerospace Engineering, Member.

(a) 1-G configuration of Ranger



(b) Neutral buoyancy configuration of Ranger

Figure 1. The two mechanical configurations of Ranger used during the testing

As a first step toward automation, this paper describes the augmentation of Ranger to autonomously identify, locate, and grasp a sampling target to be retrieved. Ranger already possesses a validated controller[2] that enables the teleoperator to smoothly drive the end-effector rather than individual manipulator joints. This work describes the integration of two technologies with the Ranger system: 1) a machine vision system to identify and locate the target to be retrieved, and 2) a trajectory planner to build and execute joint-space motions required to grasp the target and return the system to its stowed configuration. For this work, autonomous sampling was performed in both 1-G and neutral buoyancy environments. For the 1-G testing, Ranger operated in its normal configuration and a stereo camera pair was rigidly affixed to the body to provide data for the machine vision system. For the neutral buoyancy testing, Ranger operated in its wide-body configuration and an existing stereo camera pair mounted inside the body's electronics housing provided the data for the vision system. For both testing environments only a single 8-DOF dexterous manipulator was active and the positioning leg was locked to hold the body in place. However, during neutral buoyancy testing, a second 8-DOF dexterous manipulator was used to provide additional camera views for the operator, but was not involved in the autonomy. Once calibrated, the vision system was the exclusive means by which our sampling target, a yellow rubber duck, was accurately identified and located in the manipulator's workspace.

Below, the Ranger software architecture is first overviewed, followed by a description of the path planning and machine vision components developed in this work. Results focus on vision system accuracy when locating the sampling target and the overall system's ability to ultimately position its end-effector at the sampling target. This paper concludes with a brief summary and discussion of future extensions to further promote space robot autonomy and accomplish an ambitious mission to autonomously acquire biologic and geologic samples around deep-sea hydrothermal vents.

II. System Overview

The autonomy software was assembled from three primary components shown in Figure 2: 1) the Ranger distributed control software already in place for teleoperation,[1] 2) a vision-based target acquisition system, and 3) an autonomy control station. The reader is referred to Roderick et al[3,4] for more information on the Ranger software safety system always active during both teleoperated and autonomous modes.

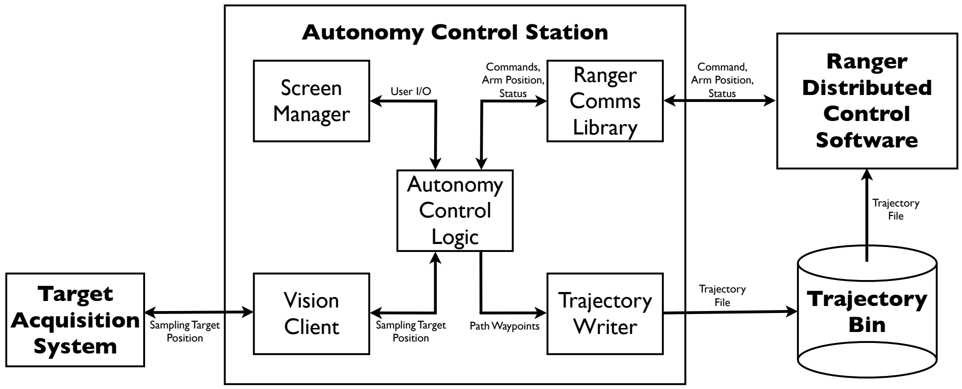


Figure 2. Autonomy software architecture

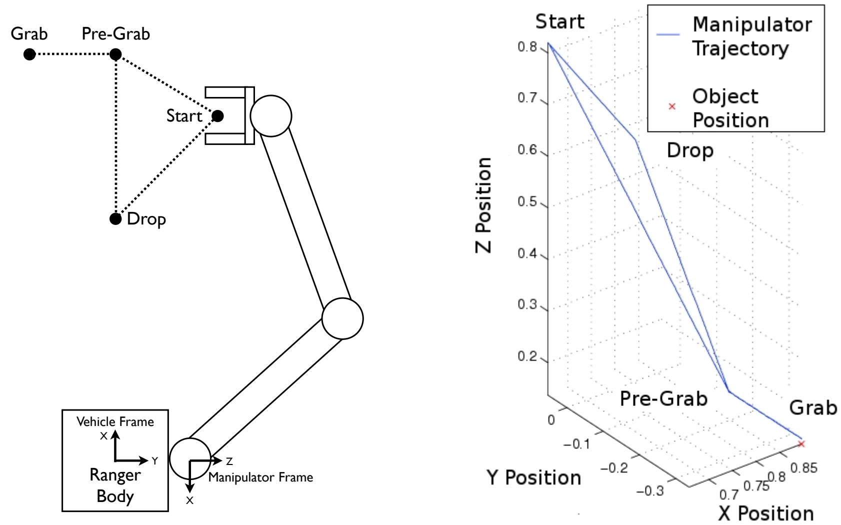
The autonomy control station directs the activity of the Ranger dexterous manipulator and the target acquisition system to enable autonomous sampling and servicing. As shown in Figure 2, the autonomy control station is built from five software modules. The screen manager handles user I/O through a simplified text-based interface considered sufficient to monitor autonomous operations. The vision client manages communication with the target acquisition system. The trajectory writer produces text-based trajectory files for commanding Cartesian-space paths to the Ranger dexterous manipulator. The Ranger communications library manages command and status data flow to and from the vehicle respectively, and the autonomy control logic coordinates the sequence of autonomy tasks. When a user prompts the system to start, the autonomy control logic requests the sampling target position from the target acquisition system through the vision client. If an object is found and is within specified workspace limits, a set of waypoints are generated to grab the object. The waypoint data is passed to the trajectory writer where the corresponding dexterous manipulator trajectory file is written and placed in a trajectory bin that the vehicle can access across the network. The autonomy control logic then commands the vehicle to load and run the trajectory file.

III. Manipulator Path Planning

Shown in Figure 3(a), a set of four waypoints is used to describe the sequence of manipulator movements for the sampling task. The manipulator begins at the "Start" waypoint and moves to the "Pre-Grab" waypoint. The "Pre-Grab" waypoint is positioned 20cm in the positive y-direction of the vehicle frame from the "Grab" waypoint. The orientation of the gripper mechanism at the "Pre-Grab" waypoint is defined such that the jaws are parallel to the vehicle frame y-axis. The manipulator remains stationary at the "Pre-Grab" waypoint while the grippers are opened. Once the grippers are fully open, the manipulator moves to the "Grab" waypoint. The "Grab" waypoint is located at the sampling target position determined by the target acquisition system. When the manipulator is at the "Grab" waypoint the arm stops and closes the grippers to grasp the sampling target. The manipulator then proceeds to the "Drop" waypoint where the grippers are opened and the sampling target is released into a storage bin. The sequence completes by moving back to the "Start" waypoint and closing the grippers. A plot of the end-effector trajectory for a typical execution sequence is shown in Figure 3(b).

An 8-DOF Cartesian-space trajectory planner was developed for Ranger to enable autonomous manipulation. Previously existing operational modes for Ranger included a resolved-rate Cartesian-space mode for teleoperation using a set of hand controllers, a joint-by-joint mode for commanding individual joint movements, and a joint-space trajectory mode for scripting joint movement sequences. These existing modes of operation were designed for teleoperation, however, Cartesian-based programmatic control was required for autonomous system behavior.

The 8-DOF Cartesian-space trajectory planner allows high-level path specification though a set of coarse Cartesian waypoints describing end-effector positions and orientations (poses). The current implementation uses linear interpolation without blending between waypoints to create each path segment.[5] Parabolic blending is currently being implemented in order to reduce long-term wear on the system and enable faster



(a) Autonomous waypoints

(b) Autonomous trajectory data (vehicle frame coords in meters)

Figure 3. Visualizations of manipulator waypoint information

movement. Cartesian end-effector waypoint pose sequences are specified in a text-based trajectory file. The end-effector poses are specified by an x-position, y-position, z -position, roll-angle, pitch-angle, yaw-angle, shoulder-elbow-wrist (SEW) angle, and a hand-roll angle. SEW and hand-roll angles are required because of the redundancy in the 8-DOF dexterous manipulator. Manipulator speed is specified by time intervals between each waypoint in the sequence. An alternate mode allows for specification of individual joint movements in a desired amount of time. This mode is used to open and close the gripper mechanism.

Execution of the Cartesian-space trajectory plan requires two steps. First, the vehicle loads the trajectory file. Here, checks on syntax are performed as well as some initial bounds checks on the data. Second, Ranger runs the trajectory file. Figure 4 shows a typical path. Path generation occurs in real time during each control cycle. For each segment in the trajectory sequence, a vector $\partial\mathbf{X}/\partial\mathbf{t}$ is calculated based on the starting end-effector pose and the desired waypoint pose. The vector specifies the change in the end-effector pose per unit time over the segment. Then, each control cycle for the currently executing segment updates the Cartesian end-effector pose as follows:

$$\mathbf{X_{new}} = \mathbf{X_0} + \mathbf{t_n}\frac{\partial\mathbf{X}}{\partial\mathbf{t}} \tag{1}$$

where $\mathbf{X_{new}}$ is the new 8-DOF Cartesian pose of the end-effector for the current control cycle, $\mathbf{X_0}$ is the starting end-effector pose for the segment, and t_n is the time of the current control cycle. Then, $\mathbf{X_{new}}$ is passed through a Newton-Raphson iterative inverse kinematic solver[5] to compute the corresponding joint positions. Finally, the new joint positions are commanded to each joint where local PD controllers close the loop on each joint's position.

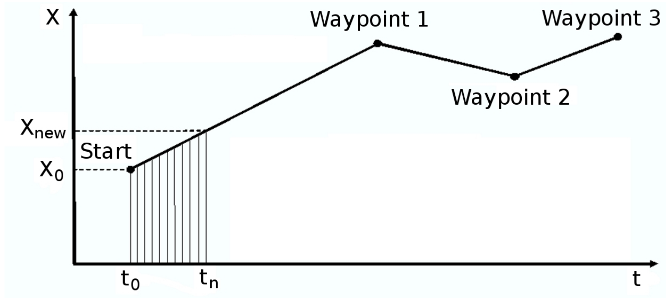


Figure 4. Path generation

IV. Vision System Calibration

A. Camera Calibration

As with any stereo vision system there are two calibration processes that must be performed to fully define the system's parameters - an intrinsic calibration for each camera and an extrinsic calibration between the two cameras. The Camera Calibration Toolbox for Matlab[6] was used to perform both of these calibration procedures.

First, pictures were taken of a checkerboard pattern using both cameras. After picking out corners of the pattern for each camera separately, the software determines the intrinsic calibration parameters: focal length, principal point, skew coefficient, and distortion coefficients. The next step is to match the corresponding checkerboard images from each camera to determine the extrinsic parameters of the stereo system. The Matlab toolbox performs this calibration automatically when given the appropriate images pairs. These parameters define the relative orientation and offset of the two cameras, expressed as a rotation matrix and translation vector. This information allows a stereo triangulation procedure to calculate the depth, Z, of points in the field of view of both cameras in addition to the two dimensional planar X, Y values.

This entire procedure was repeated for both 1-G and neutral buoyancy testing environments due to the use of different camera hardware in addition to the change in optical properties of the environment. Although the same model of camera was used in each test, Sony XC-999, Ranger's boresight cameras had poorer picture quality due to older age and a harsher work environment. This caused the calibration for the neutral buoyancy testing to be less accurate, although the underwater environment itself has excellent visual clarity.

B. Vision System to Vehicle Registration

In order to make use of the vision system's data, the relationship between the vision system's coordinate frame and Ranger's manipulator coordinate frame must be determined. Registration is the process of determining this relationship, which enables the transformation of vision system data into the manipulator's coordinate frame. Manipulator paths can then be planned based on this data. The relationship between the two coordinate frames can be expressed as a homogeneous transformation, which consists of a rotation and a translation.[5] Over the course of our testing, three different methods were used to determine the transformation in an attempt to improve the system's performance.

For the 1-G testing, a tape measure was used to determine the translation between the vision system's coordinate frame and Ranger's manipulator coordinate frame. The relative orientation was assumed based on the mounting arrangement. After a few tests it was clear that a constant rotation offset was causing positioning errors and resulting in the manipulator missing the sampling target. We then manually added small angular corrections into the transformation until the manipulator was consistently grabbing the sampling target at many different locations within the manipulator's workspace.

For the neutral buoyancy testing, Ranger's wide-body configuration prohibited accurate determination of the transformation using only a tape measure. This was mainly due to the larger distance between the coordinate frames and limited access to the vision system's cameras inside the electronics housing. As an alternative, we constructed a small checkerboard that could be grasped by Ranger's manipulator and held within the field-of-view (FOV) of the vision system as shown in Figure 5.

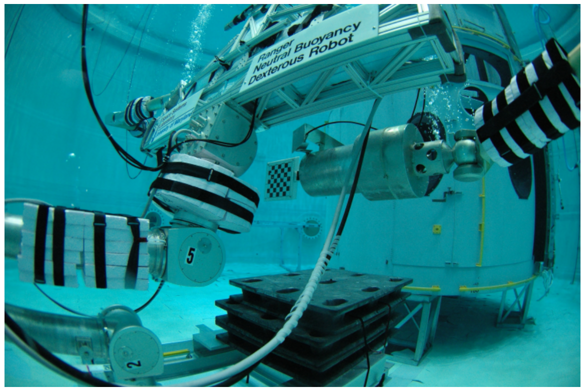
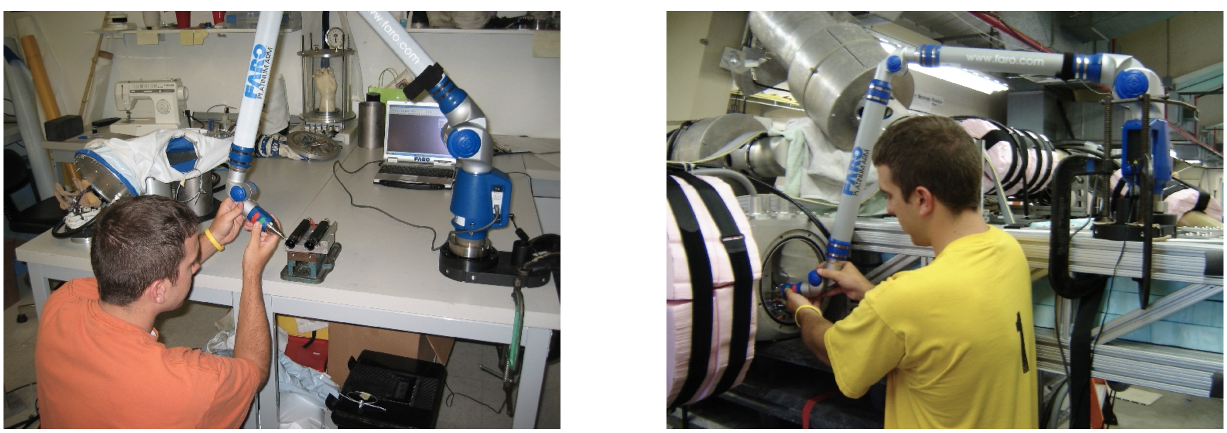


Figure 5. Registration using a checkerboard

Approximately 10 images were collected using the vision system with the checkerboard in different poses. The corresponding pose of the manipulator for each of these images was also recorded using the manipulator's encoders. Eight intersection points on the checkerboard were designated and 5 representative images were chosen producing 40 points total. The vision system's coordinates of the points were determined by manually clicking on the designated points in each image and then determining the corresponding 3D coordinate using stereo triangulation. The coordinates of the points were also determined in the manipulator's coordinate frame by using the collected end-effector pose data and knowledge of the checkerboard's dimensions. An iterative least squares algorithm was then applied to the two data sets to determine the transformation.

System level testing with the transformation determined from the small checkerboard registration method revealed that our transformation was not sufficiently accurate. This is not surprising since this method introduces errors due to the vision system's inaccuracies, which are suspected to be on the order of 5cm. To combat this, we decided to use a portable coordinate measurement machine (CMM) to determine the transformation. The device we used has a manufacturer specified accuracy of 0.0125mm, which is more than sufficient for our needs. As mentioned previously, the vision system's stereo camera pair is mounted inside the electronics box, which makes the cameras hard to measure. To aide the measurement procedure, we determined the transformation in two steps as shown in Figure 6.



(a) Vision system to bracket measurements

(b) Bracket to manipulator measurements

Figure 6. Using the portable CMM to determine appropriate transformations

First, the stereo cameras and the bracket that they are mounted to were removed from the electronics box. Designated points were measured on the cameras and the bracket and the transformation between the vision system's coordinate frame and the bracket was determined. Second, the stereo cameras and bracket were mounted back inside the electronics box and points were measured on the bracket and the manipulator's base to determine the transformation between the bracket and Ranger's manipulator coordinate frame. Combining these two transformations then gave the final transformation between the vision system's coordinate frame and the manipulator's coordinate frame.

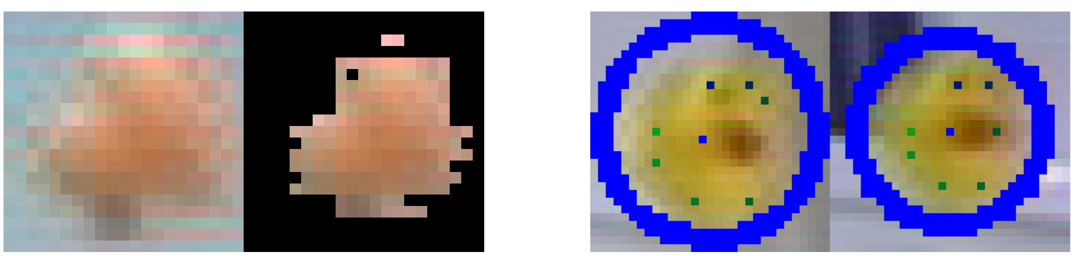
V. Target Acquisition System

The software for recording images and subsequently locating and triangulating consists of four major steps. First is simply to record the images from a live camera feed. This was done through NTSC frame grabbers (FlashBusMV) which recorded fairly low quality images at 640x480 resolution with 24bit color depth. Intel's Open Source Computer Vision Library[7] is utilized to handle the C++ image processing. The three-step target acquisition process is detailed in the following sections: 1) object identification, 2) feature matching, and 3) target localization.

A. Object Identification Filter

The first step is to filter the images based on the pixel properties of the different targets to try to pick out only the targets of interest. For the testing discussed in this paper, the filtering of the original images was based mainly on RGB color data. A separate program was written to manually adjust the filter while viewing sample images and once an appropriate filter was determined, the color values were exported for use in the main target acquisition software. Due to the difference between the two cameras in the neutral buoyancy testing a different filter was used for each camera to maximize accuracy. Figure 7(a) shows an example of the target before and after filtering. Note the low resolution and small area of the targets.

After doing the simple RGB filtering there is still a substantial amount of noise present in the image. A recursive process searches through the filtered image for pixels that passed through the filter. After locating a pixel that has remained "on," the size of the feature is determined by another recursive procedure that counts all connected "on" pixels. If the size is either above a specified threshold (feature is too large) or below a specified threshold (image noise) that feature is ignored. At this point only appropriately sized targets with proper color remain. Each of these features are then recorded based on size and location for the next step.



(a) Application of object filter

(b) Point matching of corresponding features

Figure 7. Two stages of the target acquisition system

B. Feature Matching

Stereo triangulation requires corresponding target features to be matched between image pairs. While this matching is trivial when the only feature present in each image is the correct match, there are still tests that must be done to ensure this, as well as a more complicated testing procedure for matching features when multiple targets are located in the stereo system FOV. The current feature matching algorithm compares the size of the features between the images in addition to the shape and orientation of the feature determined by the shape detection algorithm detailed in the next paragraph. By comparing the results from these methods and minimizing the differences, the "best match" can be determined for each feature. In addition to these criteria, the software ensures that matched features must physically lie in the field of view of the cameras; if there were two of the exact same target in different locations in each image, theoretically the "best match" might actually not be the correct match, but due to geometric constraints that match could be thrown out. Figure 8 shows testing in the 1-G environment with multiple targets in camera view. All targets are properly matched as indicated by the colored circles.

The shape detection algorithm chooses a set of nine points from the feature – eight points around the edge of the feature, which consist of the various combinations of maximum/minimum X and Y values in addition to the centroid. Figure 7(b) shows the points matched between corresponding features; the different colored points denote matches. By determining the unit vector and magnitude from the centroid of the feature to each of the eight outer points a mathematical "shape" is calculated which can be used for comparison between images. By minimizing the differences in each of the quantities calculated here, ideally targets with the same shape and orientation will be matched correctly between paired images.

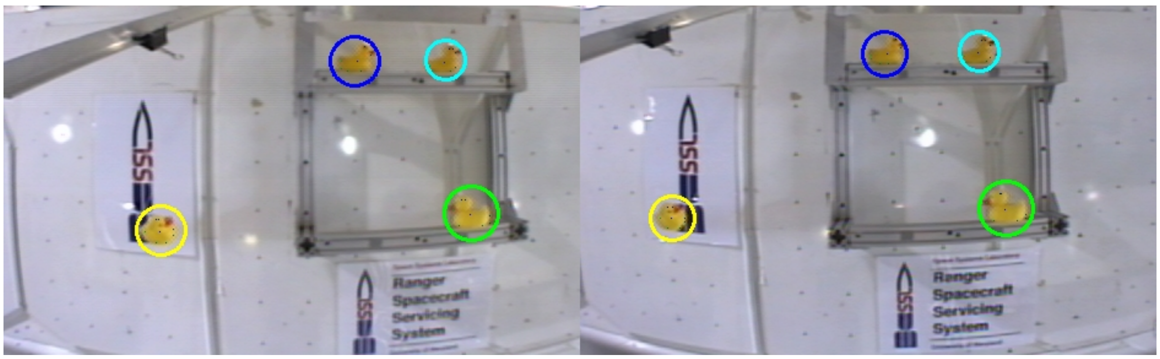


Figure 8. Multiple features matched correctly between paired images

C. Target Localization

The final step of the target acquisition software is to calculate the 3-D location of the target so that it can be retrieved. Once again, an algorithm from the Camera Calibration Toolbox for Matlab[6] was utilized. There is a *stereo_triangulation* method in this toolbox which the authors converted into C code for integration with the system. Initially, all nine points from the feature matching section were matched (via a similar point-correspondence algorithm) and then the results averaged to get the target position. The current software compares each of the eight outer points with each of the points on the matched feature from the other image and minimizes differences in the location from the centroid to determine which points match best. Usually this results in a one-to-one match, but sometimes, due to the target boundary blending with the background, some mismatches would occur. Pixels within this boundary zone confuse the filter and sometimes remain while sometimes are filtered out. This loss or addition of data between matched images can cause the mentioned point mismatches. When positions for each of the corresponding point matches have been determined they are averaged to obtain the target position. Further testing utilized only the centroid point to determine position, but the precision of the system suffered slightly in this case. Once the software has determined a position for all valid targets within the stereo pair's FOV the coordinates are transferred to the autonomy control station.

VI. Experimental Results

A. Vision System Accuracy

During 1-G testing, an effort was made to produce an initial accuracy characterization of the target acquisition system. The concept was to use the manipulator as a measurement tool since its positioning accuracy was expected to be at least an order of magnitude greater than that of the vision system. A target was fixed to the wrist of the manipulator and was moved to 11 different static locations within the workspace of the manipulator and within the FOV of the vision system. Position data for the target from the target acquisition system was collected as well as manipulator pose data at each of the 11 locations. Target position data was then derived from the manipulator data, based on the registration between the target and manipulator, and compared with the data from the target acquisition system. Table 1 summarizes the results. Of the eleven trials, errors ranged from 3.8-8.0cm with an average error of 5.3cm. Note that the negative X-axis for the coordinate frame used for this data corresponds to distance (range) from the camera.

Table 1. Target acquisition system accuracy data

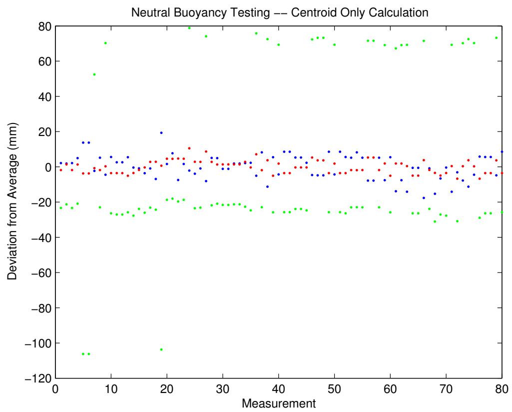
	Target Acquisition System			Vehicle Telemetry			
Test Number	X(m)	Y(m)	Z(m)	X(m)	Y(m)	Z(m)	Difference(m)
1	-0.605021	0.108046	-0.190153	-0.613912	0.123313	-0.142550	0.050776
2	-0.500795	-0.006409	0.006887	-0.487219	-0.000092	0.049443	0.045113
3	-0.480502	0.013998	-0.381687	-0.496984	0.019453	-0.347571	0.038279
4	-0.647140	-0.022447	-0.286029	-0.673603	0.009389	-0.229704	0.069902
5	-0.627976	0.013810	-0.072247	-0.624863	0.029625	-0.018874	0.055754
6	-0.717112	-0.001354	0.053585	-0.704321	0.022090	0.129046	0.080047
7	-0.731799	0.015363	-0.112521	-0.727919	0.034779	-0.039679	0.075485
8	-0.625672	-0.003226	-0.172590	-0.637247	0.004420	-0.129886	0.044901
9	-0.518175	-0.088380	-0.170819	-0.539890	-0.077760	-0.135630	0.042692
10	-0.558125	-0.185671	0.020760	-0.554598	-0.177492	0.061827	0.042022
11	-0.450209	-0.142917	-0.233936	-0.481074	-0.146701	-0.209018	0.039848

We believe there are two significant sources of error. One source is the poor registration between the vision system and the manipulator coordinate frames for the 1-G testing. Even small rotational errors cause significant positioning error at extended distances from the cameras, which is evident in tests six and seven. Additionally, recent accuracy characterization of the Ranger manipulator in its normal configuration has shown that errors up to 2.0cm are possible due to poor absolute encoder resolution which is used to determine initial joint positions before the much higher resolution incremental encoders are used. A more accurate characterization of the target acquisition system is underway using the portable CMM.

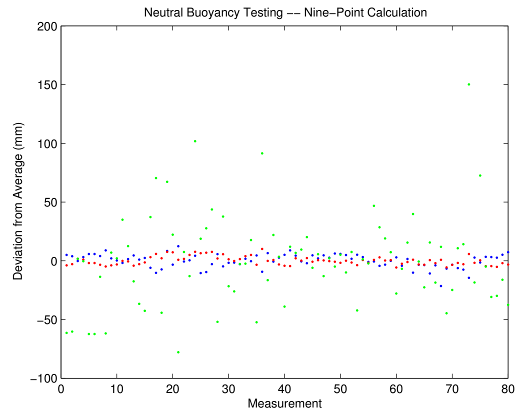
B. Vision System Precision

Without an external measurement system it is difficult to get the accuracy data presented previously, however, precision of the vision system is easily measured. In both testing environments a large number of images were taken with the target remaining stationary (500 in 1-G testing and 80 in neutral buoyancy). The results from the 1-G testing showed a much more precise system due to better camera quality: standard deviation was 2.7mm in X, 2.8mm in Y and 5.7mm in Z. Data was recorded for neutral buoyancy by averaging over all points around the target and with only the centroid. In the testing with all nine points, standard deviation in X was 5.86mm, 3.63mm in Y and 39.13mm in Z while with only centroid data standard deviation was 7.11mm in X, 3.65mm in Y and 47.87mm in Z. Figure 9 shows the neutral buoyancy data. These plots show that in the cases where only the centroid was used, the averages jump consistently between two levels. This is because the triangulation algorithm is based mainly on horizontal shift in pixel location of matching points between the two stereo images. As the amount of horizontal shift for the centroid between left and right images varies discretely, so does the calculated depth. However, on each of these levels the calculated positions are very close together. Conversely, when all nine points are used in the algorithm, the standard deviation drops significantly, yet the values are spread farther apart than on the discrete levels associated with the centroid-only calculation. During the testing, the centroid-only method was utilized, but all values were monitored so that only results from the correct level were sent to the manipulator. Removing all the data points from the incorrect levels, the standard deviation reduces to 2.80mm in the Z direction.

We believe the calculated target position jumped between those discrete levels during neutral buoyancy testing because of the dramatic increase in manipulator size. Due to workspace limitations, the target had to be placed about three times as far from the vision system: 1450mm in neutral buoyancy and only 500mm in 1-G. This caused the target pixel area to reduce significantly, thus the fixed-size boundary blending between target and background occupied a much larger percentage of feature size. In neutral buoyancy, the target filled an area of only 12x12 pixels while in 1-G it filled an area of 25x25 pixels. The boundary blending adds 2-3 pixels around the border on all sides of the target – a much more substantial percentage of the more



(a) 80 data points from NBRF test – centroid only



(b) 80 data points from NBRF test – all points

Figure 9. Plots of positions calculated during different phases of the testing

distant neutral buoyancy target. This results in less predictable filter output, which causes the triangulated location to vary more significantly. When the target was placed at approximately the same distances in 1-G and underwater environments, the results were similarly promising, indicating the underwater environment itself did not impact the vision system precision.

C. Overall System Behavior

Despite vision system inaccuracies, autonomous sampling sequences were quite effective in practice. Out of approximately 30 test runs in the 1-G environment, only two were unsuccessful in grasping the target. The first failure occurred when running Ranger in a different mode such that the tool offset from the end-effector was ignored. This caused Ranger to attempt the grab 37cm away from the duck target. The second failure resulted in improper target acquisition by the vision system. In an attempt to acquire live footage for this paper, the video feed from the left camera was split into a video recording system. Unfortunately this reduced the quality of the feed and made the left image much darker than that recorded by the right camera. The different in brightness between the two images caused the filters to incorrectly select and match corresponding points, resulting in an incorrect object position estimate. Other tests in a variety of lighting conditions were successful so long as both left and right images had similar brightness.

Neutral buoyancy testing was also quite successful considering the dramatic increase in manipulator size and increase in target distance from the vision system. After performing the accurate registration process with the portable CMM and constraining the target localization to the appropriate level of calculation, a successful target retrieval occurred for one out of seven tries. In each case where the target was missed, the manipulator was systematically too close in the depth dimension hitting the target but not grasping it with the end-effector. Small errors in the rotation aspect of the vision system to manipulator registration are suspected to have caused this error. We expect more precise calibration to remove those systematic errors. Figure 10 shows successful tests in both environments.

VII. Conclusions and Future Work

This paper has presented initial work to automate the Ranger robotic manipulator system for object acquisition and retrieval/sampling tasks. Existing Ranger control software was reused to the extent possible, enabling efforts to be focused on the addition of vision-based target acquisition and manipulator trajectory planning modules required to migrate from teleoperated to autonomous control.

The test results in both environments provided extremely promising results. Although the 1-G testing had a much higher retrieval rate than the neutral buoyancy testing, the known systematic errors can be dealt with through continued testing. More accurate knowledge of the rotation transformation between vision and

(a) 1-G – view from vision system

(b) Neutral Buoyancy – view from Ranger's left arm

Figure 10. Successful target retrieval in both testing environments

manipulator coordinate systems will decrease the error. Higher quality cameras combined with a developed filtering process will ensure that the provided position data is accurate. We are working to automate the vision system to manipulator registration process as well as automatically complete a full extrinsic calibration parameter set. Ultimately, the target filtering procedure must be extended to locate more complex targets than the yellow rubber duck.

The University of Maryland has teamed with Woods Hole Oceanographic Institute (WHOI) to develop technologies enabling autonomous planetary sample collection by merging NASA-supported robotics technologies with advanced autonomous undersea vehicles (AUVs) developed at WHOI. The Earth-based evaluation of autonomous sampling capabilities will be performed with a new, more compact manipulator that is derived from Ranger software and hardware technologies and can withstand 5000-meter seawater depths. The autonomy software described in this paper will be transferred to this undersea manipulator and tested in progressively less-controlled environments ranging from 1-g to deep sea exploration of hydrothermal vents. Vision, trajectory planning, and control technologies will require further maturation before autonomous deep-sea manipulation is practical. This research provides a controlled first step toward this ambitious exploration goal.

References

[1]Gefke, G. G., "Ranger Telerobotic Shuttle Experiment Status Report," *IEEE Intelligent Systems*, 2001.

[2]Carignan, C. and Howard, R., "A Partitioned Redundancy Management Scheme for an Eight-Joint Revolute Manipulator," *Journal of Robotic Systems*, September 2004.

[3]Roderick, S., "The Ranger Robotic Satellite Servicer and Its Autonomous Software-Based Safety System," *IEEE Intelligent Systems*, September/October 2004, pp. 13–20.

[4]Roderick, S., "An Autonomous Software Safety System for a Dexterous Space Robot," *Journal of Aerospace Computing, Information, and Communication*, Vol. 1, December 2004, pp. 564–579.

[5]Craig, J. J., *Introduction to Robotics: Mechanics and Control*, Prentice Hall, 3rd ed., 2004.

[6]Bouguet, J.-Y., "Camera Calibration Toolbox for Matlab," http://www.vision.caltech.edu/bouguetj/calib_doc, 2004.

[7]Intel, "Open Source Computer Vision Library," http://www.intel.com/research/mrl/research/opencv.