High-Accuracy 3D Sensing for Mobile Manipulation: Improving Object Detection and Door Opening

Morgan Quigley, Siddarth Batra, Stephen Gould, Ellen Klingbeil,
Quoc Le, Ashley Wellman, and Andrew Y. Ng
Computer Science Department
Stanford University
{mquigley, sidbatra, sgould, ellenrk, quocle, wellman, ang}@cs.stanford.edu

***Abstract*— High-resolution 3D scanning can improve the performance of object detection and door opening, two tasks critical to the operation of mobile manipulators in cluttered homes and workplaces. We discuss how high-resolution depth information can be combined with visual imagery to improve the performance of object detection beyond what is (currently) achievable with 2D images alone, and we present door-opening and inventory-taking experiments.**

I. INTRODUCTION

In this paper, we propose employing high-resolution 3D sensing on mobile manipulators. Just as the change from sonar-based sensing to laser-based sensing enabled drastic improvement of SLAM in mobile robotics, we propose that dramatically improving the quality of depth estimation on mobile manipulators can enable new classes of algorithms and higher levels of performance (Figure 1). In support of this idea, we present two scenarios where high-accuracy 3D data proves useful to large mobile manipulators operating in cluttered environments.

The first scenario involves object detection. In many tasks, a mobile manipulator needs to search for an object class in a cluttered environment. This problem is challenging when only visual information is given to the system: variations in background, lighting, scene structure, and object orientation exacerbate an already-difficult problem. We demonstrate that augmenting state-of-the-art computer vision techniques with high-resolution 3D information results in higher precision and recall than is currently achievable by either modality alone.

The second scenario involves manipulator trajectory planning. We demonstrate closed-loop perception and manipulation of door handles using information from both visual images and the 3D scanner. The high-resolution 3D information helps ensure that the trajectory planner keeps the manipulator clear of the door while still contacting the door handle.

We then present an application experiment which combines these capabilities to perform a simple inventory-control task. The mobile manipulator enters several offices and searches for an object class, recording the detected locations.

II. MOTIVATION AND RELATED WORK

Augmenting vision algorithms with 3D sensing has the potential to reduce some of the difficulties inherent in image-only object recognition. Prior work has shown that low-resolution depth information can improve object detection by removing object classifications which are inconsistent with training data (e.g., objects are usually not floating in the air, and some object classes are unlikely to be on the floor) [1].

Fig. 1. Several off-axis views of a raw scan of a coffee mug obtained by our scanner from 1.2 meters away. The 5mm-thick handle is prominently visible. Approximately 1000 points of the scan are on the surface of the coffee mug, despite the fact that it comprises only 5% of the horizontal field-of-view of scan.

However, if a depth sensor's noise is comparable to the size of the target object classes, it will be hard-pressed to provide more than contextual cues. The difference between a stapler and a coffee mug, for example, is only several centimeters in each dimension. Indeed, many objects designed for manipulation by human hands tend to be similarly sized and placed; thus, using depth information to distinguish

Fig. 2. Clutter makes scene understanding from only 2D visual images difficult, even in a relatively simple office environment, as many of the strong edges are not those which suggest the 3D structure of the scene.

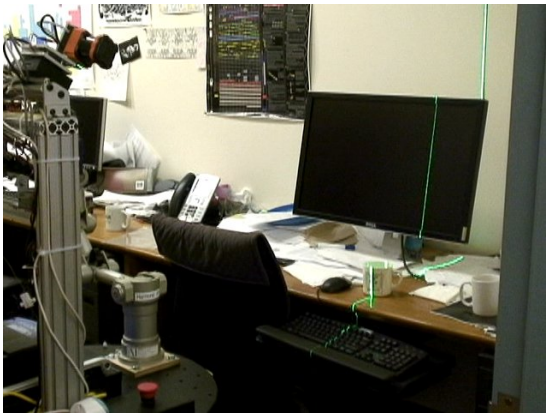

Fig. 3. A vertical (green) laser line projected by the robot at left is deformed as it strikes objects in the scene.

among them requires sub-centimeter accuracy.

Unfortunately, many current sensing technologies have noise figures on the centimeter level when measuring from 1-2 meters away. Ranging devices based on time-of-flight, for example, tend to have centimeter-level noise due to the extremely short timescales involved [16]. Additionally, time-of-flight ranging systems can introduce depth artifacts correlated with the reflectance or surface normal of the target object [15].

In contrast, the accuracy of passive stereo cameras is limited by the ability to find precise feature matches. Stereo vision can be significantly improved using global-optimization techniques[14], but the fundamental problem remains: many surfaces, particularly in artifical environments, do not possess sufficient texture to permit robust feature matching (e.g., a blank piece of paper). Efforts have recently been made to combine passive stereo with time-of-flight cameras [13], but the resulting noise figures still tend to be larger than what is achievable using a laser line scanner.

Active vision techniques use yet another approach: they project large patterns onto the scene using a video projector, and observe deformations of the patterns in a camera to infer depth [17]. Besides the difficulties inherent in overcoming ambient light simultaneously over a large area, the projected image must be at least roughly focused, and thus depth of field is limited by the optical geometry. However, this is a field of active research and great strides have been made in recent years.

This brief summary of the limitations of alternative 3D sensing modalities is bound to change with the continual progress being made in each of the respective areas of inquiry. In this paper, we seek to explore the potential benefits of highly accurate 3D data for mobile manipulators. As other 3D modalities continue to improve, their data could be used by the algorithms described in this paper. For the purposes of this study, we have built several 3D laser line triangulation systems to explore how high-quality 3D data can improve the performance of mobile manipulation.

We selected laser line triangulation because millimeter-level accuracy is readily achievable. This is on the order of accuracy we have been able to achieve in sensor-to-manipulator calibration; further increases in sensing accuracy would thus not necessarily improve manipulation performance.

Laser line scanners have proven useful in manufacturing, as is well documented both in the research literature [7] and in the marketplace [8]. They have been often used in fixed installations, where objects are placed on a rotary table in front of the scanner [9] or flow by on conveyor belts. Low-cost implementations have been designed which rely on a known background pattern instead of precision hardware [10]. Triangulation-based laser scanners have also been used on planetary rovers to model rough terrain [12], to find curbs for autonomous vehicles [2] and to model archaelogical sites and works of art [3].

Out-of-the-box triangulation systems are commercially available for imaging small objects [4]. However, many of these systems emphasize high accuracy ($<$ 0.1mm), often sacrificing depth of field. To be of most use to a mobile manipulator, the sensor needs to cover the entire workspace of the manipulator, and "extra" sensing range is helpful in determining how to move the platform so that a nearby object will enter the workspace of the manipulator.

Although triangulation-based laser scanners have been proposed for mobile manipulators in the past [11], at time of writing, we are not aware of implemented systems similar to what we describe in this paper.

III. LASER LINE SCANNING FOR ROBOTICS

Our scanner is intended to complement computer vision systems on mobile manipulators. As such, we aim to produce a depth estimate for each pixel in the scene. The resulting images can be considered as having an "extra" channel representing each pixel's depth, in addition to the usual RGB- or monochrome-intensity channels.

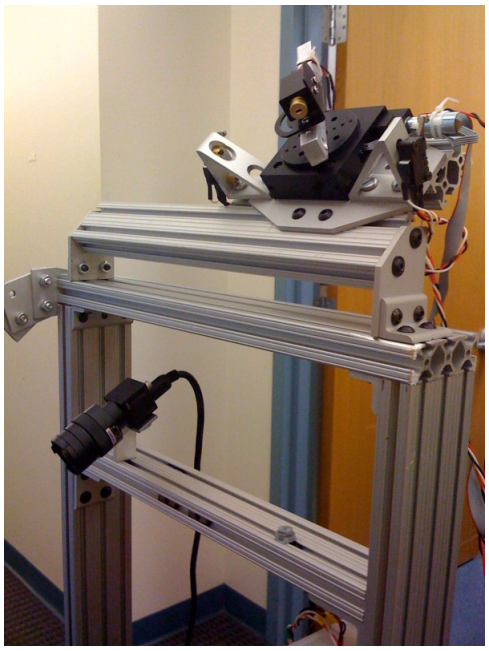

Fig. 4. The scanning hardware on the STAIR 1 robot: the laser and rotary stage are mounted in the upper-right. Images are captured by the camera in the lower-left.

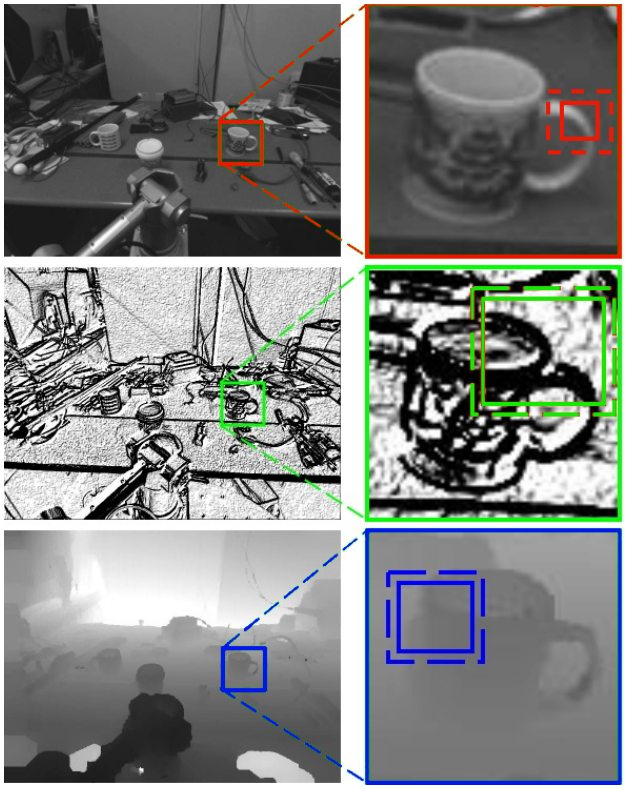

Fig. 5. Image channels considered by the patch-selection algorithm, along with the typical appearance of a coffee mug. A typical patch selected by the system is boxed on right. The larger dashed box indicates the typical 7-pixel window used when finding the maximum patch response. Top (red): intensity image. Middle (green): gradient image. Bottom (blue): depth image.

A. *Fundamentals*

The geometry of the laser-line triangulation scheme is well-studied and only repeated here for completeness. Many variants of the underlying concepts are possible. In our scanners, a rotating vertical laser line is directed into the scene. An image formed on a rigid, horizontally-offset camera shows a line which is deformed by the depth variations of the scene (Figures 3 and 4). On each scanline of the image, the centroid of the laser is detected and used to define a ray from the camera origin through the image plane and into the scene. This ray is intersected with the plane of laser light defined by the angle of the laser stage, its axis of rotation, and 3D translation from the laser stage to the camera origin. The intersection of the plane and pixel ray produces a single 3D point directly in the image frame, thus avoiding the depthmap-to-image registration problem.

The vertical angular resolution of the point cloud is limited by the vertical resolution of the camera. The horizontal resolution is determined by the the laser's rotation speed, the camera's frame rate, and the field of view. Depth resolution is determined by a variety of factors: the ratio between the horizontal resolution of the camera and the field of view, the precision of the shaft encoder on the laser stage, the ability to achieve horizontal sub-pixel interpolation, the horizontal offset between the camera and the laser, and the distance of the object from the camera.

B. *Hardware Considerations*

We acquire roughly 600 images during each scan. The camera's field of view is approximately 70 degrees, and we overscan by 10 degrees to accomodate for the depth variations of the scene. As a result, the laser line moves approximately 0.15 degrees per frame.

Our scanner currently requires 6 seconds to gather the images, which are buffered in RAM on a computer onboard the robot. Subsequent image processing and triangulation steps require an additional 4 seconds. Such a slow rate of acquisition means that the scanner cannot be used in fast-moving scenes. This is a fundamental limitation of line scanning; however, additional implementation effort could result in dramatic speedups, e.g., moving to (very) high-speed cameras and/or performing the image processing on a graphics processor (GPU).

C. *Calibration*

We used the automatic checkerboard-finding algorithm and nonlinear solver implemented in OpenCV [20] to estimate the camera intrinsics. To estimate the camera-laser extrinsics, we start by roughly measuring the translation and rotation by hand. We then scan a flat board marked with several points whose relative planar distances have been carefully measured. The locations of these points in the camera image are found and recorded. We can then quantify the calibration error: the projected 3D points should be coplanar as well as exhibit the measured distances. We image this test board from several angles to better cover the workspace of the

scanner. A numerical optimization routine is used to minimize the sum of the errors while perturbing the parameters, randomly restarting many times to explore many different local minima.

The resulting calibration holds true except at the extreme edges of the camera view. We assume this is due to effects of the lens not captured in the standard radial and tangential distortion models. Away from the edges of the image, the scanner shows errors in the 1mm range when imaging flat surfaces such as doors, desks, and walls.

To calibrate the manipulator to the scanner, we need to define the 6D transform between the manipulator's base frame and the camera frame. To accomplish this, we touch the manipulator's end effector to several points on a test board which are identifiable in the camera frame, logging the manipulator's forward-kinematics position each time. We then employ a numerical optimization routine to improve our hand-measured estimate of the 6D transform. The resulting calibration accuracy is approximately 5mm throughout the workspace of the manipulator.

IV. OBJECT DETECTION

Once the scanner has been calibrated, it can be employed to improve the performance of object detection. For many robotics applications, this is a critical subgoal of a larger task: for example, in order to grasp an object it is first necessary to detect the presence (or absence) of the target object and localize it in the workspace.

As previously mentioned, the scanner aims to produce a depth estimate for every pixel. Although the geometry results in the depth being estimated in the image plane, the information does not lie on a regular grid due to sub-pixel horizontal interpolation used to estimate the center of the laser stripe. Furthermore, some regions of the depth image will be more dense than others, depending on the direction of the surface normal and the distance to the surface. We thus resample the depthmap using bilinear interpolation to match the raster of the camera. At this point, the depthmap can be viewed as another channel in the image.

A. Sliding Windows

Sliding-window methods attempt to match a rectanglar window of the image with a collection of very small features, or "patches," stored in a probabilistic classifier.

This classifier can be viewed as a black-box which returns a high probability if the window tightly bounds an instance of the target object class, and a low probability otherwise. To perform object detection across an entire image, the window is shifted through all possible locations in the image at several spatial scales.

We use an extention of the sliding-window approach to combine information from the visual and depth channels. Similar to the state-of-the-art approach of Torralba et al. [5], the features used by the probabilistic classifier are derived from a learned "patch dictionary." Each patch is a very small rectangular subregion randomly selected from a set of hand-labeled training examples. The channels considered are the original (intensity) image, the gradient image (a transformation of the original image: edges become bright, flat regions become dark), and the depth map discussed in the previous section. The patches are drawn separately from these three channels, and probabilistically represent the visual appearance (intensity or edge pattern) or shape (depth profile) of a small region of the object class (Figure 5).

Combined, the patches give a generalized representation of the entire object class that is robust to occlusion and appearance or shape variation. When constructing the dictionary, we record the patch g, its location within the window containing the positive example w, and the channel from which it was drawn c (intensity, gradient, or depth). A *patch response* for a particular window is computed by measuring the similarity of the corresponding region within the window to the stored patch.

More formally, let the image window be represented by three channels $\{\mathcal{I}^i, \mathcal{I}^g, \mathcal{I}^d\}$ corresponding to intensity, gradient and depth, respectively. Then the patch response for patch $p = \langle g, w, c \rangle$ is

$$\max_{w'} d^c(\mathcal{I}^c, g)$$

where $d^c()$ is a similarity metric defined for each channel. To improve robustness to minor spatial variations, w' is a 7×7 pixel grid centered around the original patch location in the training set. This allows the patches to "slide" slightly within the window being tested.

We compute similarity between patches using normalized cross-correlation. For the intensity and gradient channels we normalize by subtracting the average (mean) from within the window; for the depth channel we normalize by subtracting the median depth.

B. Learning the Classifers

The preceeding discussion assumed that the classifier was already known. In this section, we discuss how the classifier is built from training data.

For each object class, we learn a binary gentle-boost classifier [6] over two-split decision stumps in these steps:

- Construct a training set by cropping positive examples and random negative windows from our training images.
- Build an initial patch dictionary by randomly sampling regions from our positive training images, and compute patch responses over our training set.
- Learn a gentle-boost classifier given these responses.
- Trim the dictionary to remove all patches that were not selected by boosting.
- Run the classifier over our training images and augment our set of negative examples with any false-positives found.
- Repeat the training process with these new negative examples to obtain the final classifier.

Since we are learning two-split decision stumps, our classifiers are able to learn correlations between visual features (intensity patterns and edges) and object shape (depth). Example patches from a coffee-mug classifier for the three

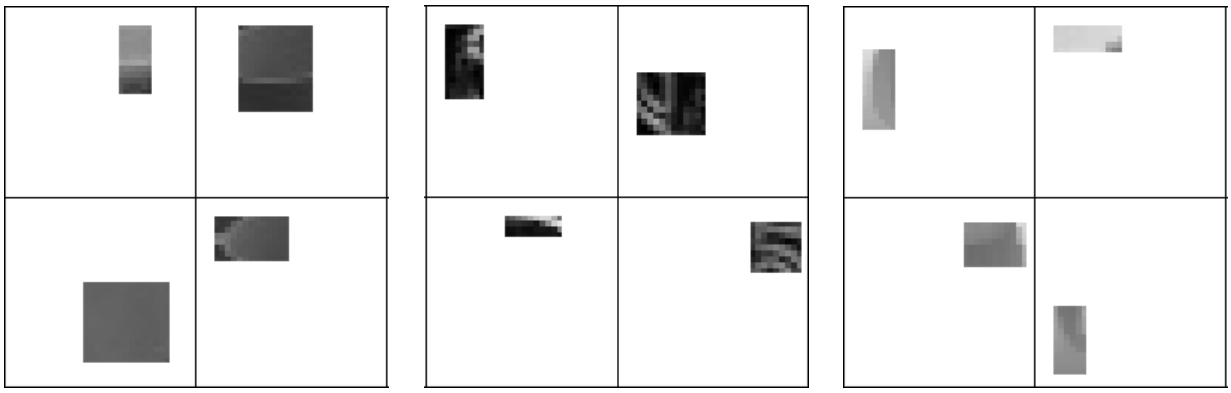

Fig. 6. Examples of localized patches from the coffee-mug dictionary. Left: Intensity patches. Middle: Gradient patches. Rigth: Depthmap patches.

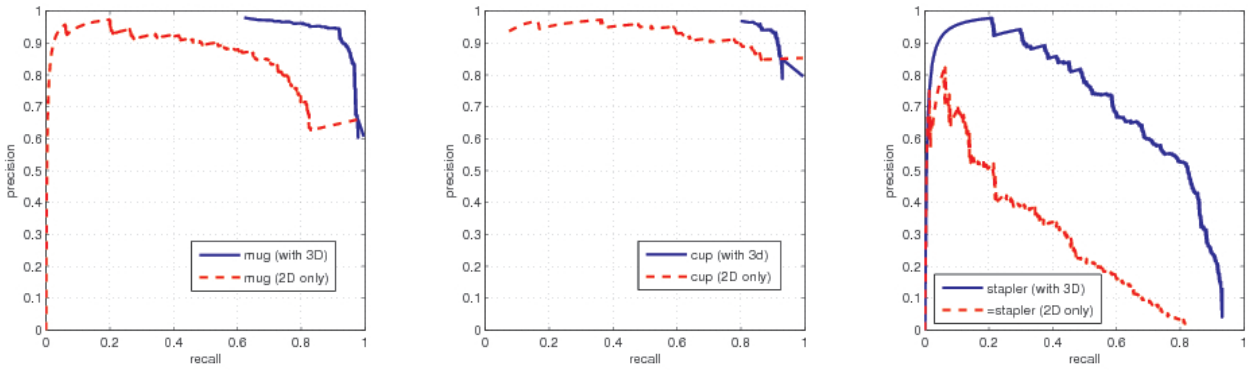

Fig. 7. Precision-recall curves for mugs (left), disposable cups (middle), and staplers (right). Blue solid curve is for our method; red dashed curve is for vision only detectors. Scores are computed at each threshold by first removing overlapping detections. A true-positive is counted if any detection overlaps with our hand-labeled groundtruth by more than 50%. Any detection that does not overlap with a groundtruth object of the correct class is considered a false-positive. Average Precision measures 11-point interpolated area under the recall vs. precision curve. Greater area under the curve is better.

image channels are shown in Figure 6. This figure is a typical representation of 12 of the approximately 50 patches selected by the algorithm.

We performed five-fold cross-validation to evaluate the performance of our detectors and compare them against state-of-the-art detectors that do not use depth information. The dataset consisted of 150 images of cluttered office scenes, with several objects in each scene. We used the same training procedure (as outlined above) for each detector and report the average performance over the hold-out sets. Results for coffee mugs, disposable cups, and staplers are shown in Figure 7 and the following table:

	Mug		**Cup**		**Stapler**	
	3D	2D	3D	2D	3D	2D
Max. F-1 Score	0.932	0.798	0.922	0.919	0.662	0.371
Average Precision	0.885	0.801	0.879	0.855	0.689	0.299

In general, the 3D information appears to help eliminate false positives. The 2D detectors seldom miss instances of their trained object class; their typical problem is instead that they can collect a variety of disparate cues from shadows or unrelated objects that together match enough of the localized patches that the sliding-window detector considers it a high-probability detection. The 3D information can help in this regard: we observe that the training process often selects relatively large, uniform depth patches. Effectively, this associates higher probabilities to windows which tightly bound a single object rather than a collection of several disparate objects. Since we do not normalize for the depth variation inside a patch, only for its median, the depth patches also encode a measure of the absolute size of an object. These depth cues are not explicitly expressed in the visual-light image, and as is common in machine learning systems, presenting a richer set of features to the classifier helps boost performance.

V. DOOR OPENING

For many tasks, mobile manipulators operating in home and office environments need to open and pass through doors. For example, at the end of a workday a typical office building will have tens or hundreds of closed doors that must be opened if the robot is to clean the building or search for an item. The ability to open a door thus needs to be another primitive in the robot's navigation toolbox, alongside path planning and localization. We summarize our door-opening system to emphasize the utility of high-resolution 3D sensing for mobile manipulation.

Fig. 8. After localizing the door handle in the 3D point cloud, the robot can plan a path to the handle and open the door.

Fig. 9. Detecting coffee mugs in cluttered environments. The detector correctly ignored the paper cup to the right of the coffee mug.

Door opening requires manipulating the door handle without colliding with the door. The operation does not allow more than a centimeter or two of positioning error, as the end effector is continually in close proximity to the (rigid) door. Thus the door-opening task, like any grasping task where target objects are identified in a camera, tests not only sensing accuracy but also the calibration between the sensing system and the manipulator.

Our system uses a hand-annotated map which marks the locations of doors. If the robot needs to pass through one of the marked doorways, it uses the triangulation-based laser scanner described in this paper to scan the door. From this scan, it uses a classifier trained on hundreds of door handles to localize the handle and classify the door as right-handed or left-handed. The robot then drives to within manipulator reach of the door handle, plans a path to the edge of the handle, and presses on the handle to unlatch it (Figure 8). Once the door is unlatched and partially opened, the robot is able to drive through the door by pushing it fully open as its chassis (slowly) comes into contact with the now-unlatched door.

High-resolution point clouds assist in planning collision-free manipulator paths to the door handle. Some sensing modalities effectively "low-pass" the depth map as part of the sensing process. In contrast, the active triangulation process does not smooth out depth discontinuities, such as those between the door handle and the door immediately behind it. As a result, the door handle stands out sharply in the 3D data, making path planning and recognition easier.

VI. INVENTORY-CONTROL EXPERIMENT

To evaluate the utility of these two uses of the laser-line triangulation scanner on our mobile manipulator, we combined the object-detection and door-opening algorithms to form an inventory-taking experiment. Such a system could be envisioned in a future home, perhaps cataloging the locations of every object in the house at night so that the robot could instantly respond to human queries about the location of commonly-misplaced objects. Workplace applications could include inventory-taking in retail stores, safety inspections in industry, or location verification of movable equipment in, e.g., hospitals.

In our system, a high-level planner sequences a standard 2D navigation stack, the door-opening system, and the object-detection system, which together allow the robot to take an inventory of an object class in a cluttered office building with closed (but unlocked) doors. Our system runs on the ROS software framework [18]. A world map was built offline using the GMapping SLAM toolkit [19] and logged data from the robot's SICK LIDAR and Segway odometry. The resulting map was hand-annotated to mark the locations of doors and desks. The runtime navigation stack is descended from the Player localization and planning modules, which perform particle-filter localization and unified object-avoidance and goal-seeking path planning.

When necessary, control switches to the door-opening system discussed in the previous section, after which motor control is returned to the 2D navigation stack.

A sample run of the inventory-gathering system is shown in Figure 10. During this run, there were 25 coffee mugs spread in the search area. The 3D-enhanced object detector found 24 of them, without any false positives. Our image-only detector was only able to find 15 of them, and it found 19 false positives. During the experiment, we also searched the scans for disposable paper cups. The mug-inventory results and the cup-inventory results are compared against ground truth in the following tables for both the integrated 3D detectors and the 2D-only detectors.

3D-Enhanced Detectors

OBJECT	COUNT	HIT	ERROR	RECALL	PREC.
Mug	25	24	0	0.96	1.00
Cup	10	8	2	0.8	0.8

2D-Only Detectors

OBJECT	COUNT	HIT	ERROR	RECALL	PREC.
Mug	25	15	19	0.6	0.441
Cup	10	8	4	0.8	0.67

Fig. 10. The inventory-gathering experiment required autonomous navigation (green track), autonomous door opening, and 20 laser scans of desks in the four offices shown above. The robot position at each scan is shown by the red circles, and the field-of-view of each scan is indicated by the yellow triangles. The locations of the detected coffee mugs are indicated by the orange circles. This figure was entirely automatically generated, using the SLAM output for the map and the localization log for the robot track and sensing positions, which allow the coffee-mug detections to be transformed into the global map frame.

VII. CONCLUSIONS AND FUTURE WORK

As shown by the PR curves obtained when using the 3D information versus 2D alone, incorporating high-quality 3D information into the sensing scheme of a mobile manipulator can increase its robustness when operating in a cluttered environment. The door-opening task shows that high-quality 3D data can help accomplish motion planning by accurately sensing the immediate vicinity of the robot.

We intend to continue increasing the speed of our 3D triangulation system by moving to ever-faster camera frame rates. We also intend to explore other modalities of obtaining high-accuracy 3D data, and quantify the performance improvement provided by various depth sensors.

REFERENCES

[1] S. Gould, P. Baumstarck, M. Quigley, A. Y. Ng, and D. Koller, Integrating Visual and Range Data for Robotic Object Detection, *European Conference on Computer Vision*, 2008

[2] C. Mertz, J. Kozar, J. R. Miller, C. Thorpe, Eye-safe Laser Line Striper for Outside Use, *IEEE Intelligent Vehicle Symposium*, December 2001

[3] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, D. Fulk, The Digital Michelangelo Project: 3D Scanning of Large Statues, *SIGGRAPH*, 2000

[4] http://www.nextengine.com

[5] A. Torralba, K. Murphy, W. Freeman, Sharing Visual Features for Multiclass and Multiview Object Detection, *NIPS*, 2007

[6] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, *Technical report, Dept. of Statistics, Stanford University*, 1998

[7] J. Li, J. Zhu, Y. Guo, X. Lin, K. Duan, Y. Wang, Q. Tang, Calibration of a Portable laser 3-D Scanner used by a robot and its use in Measurement, *Optical Engineering* 47 (1), January 2008

[8] Metris, among other companies, offers precision laser line scanners for manufacturing: `http://www.metris.com/products/robot_scanners/k-robot/`

[9] J. Jezouin, P. Saint-Marc, G. Medioni, Building an Accurate Range Finder with off-the-shelf Components, *Proceedings of CVPR* 1988, p.195-201

[10] The "David Laserscanner" is an example of this technique: `http://www.david-laserscanner.com`

[11] K. Nagatani, S. Yuta, Designing a Behavior to Open a Door and to Pass Through a Doorway using a Mobile Robot Equipped with a Manipulator, *Proceedings of IROS* 1994, p. 847-853.

[12] L. Matthies, T. Balch, B. Wilcox, Fast Optical Hazard Detection for Planetary Rovers using Multiple Spot Laser Triangulation, *ICRA* 1997.

[13] J. Zhu, L. Wang, R. Yang, J. Davis, Fusion of Time-of-Flight Depth and Stereo for High Accuracy Depth Maps, *Proceedings of CVPR*, 2008.

[14] J. Sun, N. Zheng, H. Shum, Stereo Matching Using Belief Propagation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 (7), July 2003.

[15] D. Falie, V. Buzuloiu, Noise Characteristics of 3D Time-of-Flight Cameras, *International Symposium on Signals, Circuits, and Systems*, 2007.

[16] S. May, B. Werner, H. Surmann, K. Pervolz, 3D Time-of-Flight Cameras for Mobile Robotics, *IROS*, 2006.

[17] S. Zhang, P. Huang, High-Resolution, Real-Time Three-Dimensional Shape Measurement, *Optical Engineering*, 45 (12), December 2006.

[18] The ROS (Robot Operating System) framework is an open-source, peer-to-peer, cross-platform message-passing system being jointly developed by Stanford University and Willow Garage. The ROS distribution wraps many popular code bases, such as OpenCV, GMapping, the navigation stack from Player, the Stage and Gazebo simulators, and provides drivers to various pieces of robotics hardware. ROS is available on Sourceforge. Documentation is available at `http://pr.willowgarage.com/wiki/ROS`

[19] G. Grisetti, C. Stachniss, W. Burgard, Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters, *IEEE Transactions on Robotics*, 2006.

[20] `http://opencvlibrary.sf.net`