The AAAI 2017 Spring Symposium on
Interactive Multisensory Object Perception for Embodied Agents
Technical Report SS-17-05

Merging Local and Global 3D Perception Using Contact Sensing

Rebecca Cox and Nikolaus Correll
rebecca.e.cox@colorado.edu, nikolaus.correll@colorado.edu
Department of Computer Science, University of Colorado
Boulder, CO 80309, USA

Abstract

This paper presents our ongoing work towards fusing RGB-D images with data from contact and proximity sensors embedded in a robotic hand for improved object perception, recognition and manipulation. Optical depth information from multiple sensors is often inaccurate and inconsistent. These problems arise from problems with sensor calibration, but also occlusion of objects by other objects or the robot arm itself. In this paper, we propose to combine global pose information from RGB-D sensing with local proximity sensing during approach. Here, we use contact information based on a novel contact sensor and additional pose information provided by the arm's pose.

Perception is a necessary component in grasping and manipulation that continues to be a challenge in robotics. Both 3D perception and tactile sensing have been explored, and both might be sufficient means on their own and complement each other (Patel et al. 2017). Accuracy in object recognition depends heavily on the accuracy of calibration, successful point cloud segmentation, lighting conditions, and whether objects are occluded or not. Tactile sensing (Hsiao, Kaelbling, and Lozano-Pérez 2011) can be much more accurate as the kinematics of a robot arm are usually well known and its encoders are precise, but requires active exploration of the environment.

In this paper, we describe our ongoing efforts on fusing 3D perception, in-hand proximity sensing, and touch into a single representation. Here, the key ideas are that tactile sensing can provide data-rich 3D representations with accuracy and proximity sensing can seamlessly bridge between the two sensing modalities, allowing us to gather data without disturbing the object's pose.

Using tactile sensing in perception has been widely explored, for example to plan motions to learn more about the geometry of an object or match it against a known 3D model (Hsiao, Kaelbling, and Lozano-Pérez 2011; Dang and Allen 2014; Saut et al. 2014; Ma et al. 2015). Yet, little attention has been devoted on fusing proximity and depth information obtained from different means using tactile sensing to provide a common reference frame. Such refined models could then be used to improve grasp planning.

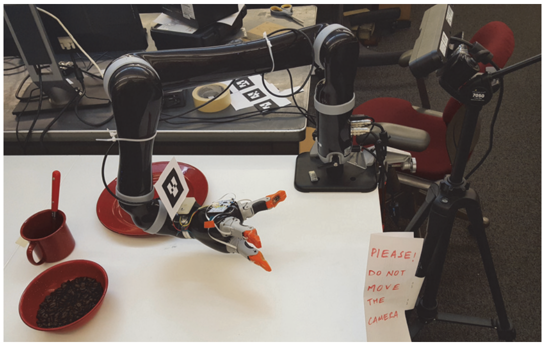

Figure 1: A typical manipulation pipeline combining multiple sensor streams. The fingers on the Jaco arm from Kinova contain proximity and force sensors and an Asus Xtion is used for depth and RGB imaging.

We are combining an Asus Xtion depth sensor with Robotic Materials' tactile sensors for the Kinova three-fingered hand and Rethink Robotics Baxter. We use the Point Cloud Library (PCL) to perform object detection from the raw point cloud. Our experimental setup is shown in Figures 1 and 2. We have built a perception pipeline that performs reasonably well at correctly labeling known objects from a small data set of a dozen tabletop objects. However, for pose estimation, accuracy and reliability remains a challenge, especially when objects are placed in a cluttered environment or have partially occluded views. Grasping objects with hidden features such as a handle on a cup not seen by the camera (Figure 2) can cause the gripper to collide with and bump the object unintentionally.

The work described here is "work-in-progress", which we hope to share with participants of the workshop. In the remainder of this paper, we describe the two sensing modalities, a 3D object recognition pipeline and the combined proximity and force sensor, as well as observations and lessons from preliminary efforts on how to fuse the information of the two.

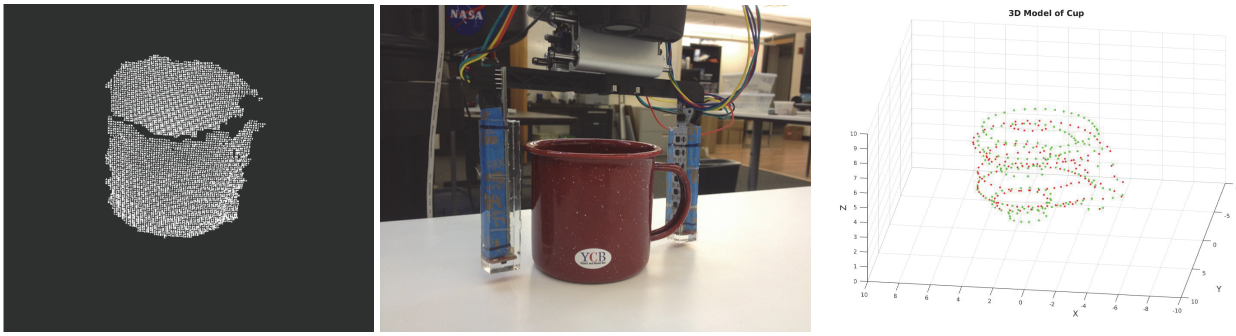

Figure 2: All three images feature the same cup from the YCB data set(Calli et al. 2015) with the handle on the right side. Left image: Segmented point cloud of the cup from the depth camera. Although the handle is in full view, it is not reflecting enough points to be registered in the point cloud. Middle image: Proximity sensors on Baxter grippers being rotated around a cup to capture 3D scan. Right image: Plot of those proximity measurements build a crude 3D model from (Patel and Correll 2016). The two different colors indicate which side of the gripper the measurements came from. The handle of the cup is clearly shown in this scan and can be used to fill in missing information in the first image.

Object Recognition and Segmentation from RGB-D data

To provide simple object detection, we built a perception pipeline using PCL that takes a raw point cloud and RGB image as input and publishes found objects and their poses. Using PCL's built in feature detection libraries, we are able to successfully perform object labeling on individual objects, but don't always have the information available in a single frame to be able to estimate object pose accurately, especially in a cluttered environment with occluded views. In such cases, the center-of-mass of the point cloud is not congruent with the actual center, making grasping difficult. Worse, key geometries that are important for grasping and manipulation are hidden and might not be inferred from partial models.

While the approach described here is basic, we believe its challenges and limitations to be ubiquitous in manipulation, even when using more advanced methods for perception.

Object Segmentation

The first part of the pipeline involves removing noise and narrowing down the point cloud to regions that contain the objects. We can assume that the objects will always be placed on a table or flat surface that will fill a large portion of the field of view in the camera. For this reason, the initial step is to find the flat surface and extract it from the point cloud using Random Sample Consensus (RANSAC), a simple method of separating inliers of the 2D plane from everything else in the cloud.

The remaining point cloud is clustered into smaller objects using a nearest neighbor approach. This is calculated by taking a sphere of radius r_s around every point in the cloud and grouping points contained in the same sphere. Objects that are placed further than the distance r_s from each other will be in different neighborhoods and will get processed separately. Objects that are stacked or within the spherical radius become clustered into a single object and require further segmentation techniques described in the next section.

We finally use color to further discriminate between different objects with overlapping point clouds using the approach described in (Zhana, Liangb, and Xiaoa 2009). An example segmentation is shown in Figure 3.

Correspondence Matching

After the objects have been segmented, we individually analyze each one and compare it against the known data set. Due to the redundancy in point cloud information of a single object, the objects are down-sampled to their keypoints. These keypoints are sparse enough to make processing much more efficient while still containing enough information for surface reconstruction. 3D descriptors from (Tombari, Salti, and Di Stefano 2010) are calculated from these keypoints and used in searching for correspondence with an object from the data set. When enough matching descriptors are found, the object is labeled and a pose is estimated based on the translations from matching descriptors. This approach is described in more detail in (Patel et al. 2017).

Tracking Found Objects

The final step in the perception pipeline aims to improve object recognition during run time by decreasing the likelihood that an object found previously will be lost due to two main reasons: objects being moved or occluded by a robotic arm during manipulation and noise in a point cloud that results in not enough correspondences to be found that were observed previously. When an object is found, we record a geometric centroid and a timestamp. In the likely case that in the next few seconds an unidentified object is found to have a near centroid, but failed correspondence matching, then it is assumed to be the same object as identified before. We found that correctly identified objects often lost their label every few frames, even without moving the object. Small fluctuations in the received point cloud and down-sampling reduce the number of matched descriptors found. Similarly, descriptor matching can fail due to the robotic arm partially occluding the view of an object during manipulation. Keeping track of found objects in this way greatly reduced false negatives seen.

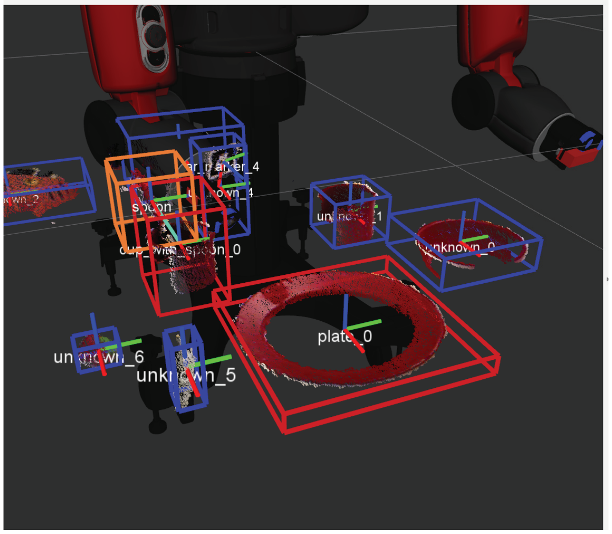

Figure 3: Segmented output from a table top manipulation scene using the Baxter robot.

Proximity and Force Sensors

We use combined proximity, contact and force sensors described in (Patel and Correll 2016) to get proximity measurements from the robotic hand to an object with a range of up to 10 cm. This information is used to improve gripper alignment during grasping and determine when contact with objects is made, which we describe in (Patel, Alastuey, and Correll 2016). The sensors are available commercially[1] and easy to integrate with existing hardware. They consists of a digital infrared distance sensor that emits infrared and measures proximity. For protection from abrasion and to allow light to pass through, they are embedded in a soft transparent polymer, which doubles as a spring for force measurements based on Hookes law. When a force is applied on the polymer, the force on the object is derived from calculating the deformation that is observed in the infrared reflection measurement. Due to an inflection point in the sensor signal at contact, contact can be robustly estimated independently of the surface reflectivity.

Using an array of sensors, we can combine proximity measurements to form a crude 3D model. As shown in Figure 2, the Baxter grippers were placed around the cup and rotated to form a 3D scan of the entire surface. Each sensor is at a defined location on the gripper. This knowledge, combined with the proximity measurement, allowed us to form the 3D model shown. Although crude, the location of the handle within the image is obvious.

Fusing Sensor Data using Contact Sensing

Contact sensing has already shown to be beneficial in our experiments during grasping by providing immediate feedback of the object's distance relative to the fingers. We know when we are within close proximity to our object when the sensor value has crossed a set threshold. Similarly, contact is determined when the value has crossed a higher threshold. Regardless of surface texture, color, and reflectivity, contact detection is consistently accurate, making it the most reliable sensory data in our system.

We show in Figure 4 that the models produced from contact are much cleaner than those obtained from proximity information and can be combined with the point cloud from the depth sensor for more complete 3D reconstruction of objects. This sensory data provides a feedback loop for our system that we plan to use to improve calibration of the finger sensors, improve camera alignment and calibration, and fuse the point clouds for better object pose estimation. Specifically, Figure 4 clearly demonstrates the large offset between the depth data, the overlaid RGB image, and the point cloud provided from tactile information. We also note that tactile information reveals additional information that is not available from the RGB-D camera, namely information from the top of the stair case (pink and blue dots resulting from touch by the fingertips), and information from the backside of the staircase (cyan dots). Just as pink and blue represent the touch from one of the two fingertips, the green and cyan represent touch from inside either the left or right finger.

Ongoing Work

To fuse the models together, we propose using contact detection from the sensors together with proprioception information from the robotic arm, which we believe to be more accurate than either optical perception system. In order to determine correspondence between the different point clouds, we plan to use similar 3D feature descriptors as used during registration to find correspondences between RGB-D and tactile point clouds. Once an initial correspondence is made, we plan to use the ICP algorithm to improve the alignment. Although standard in object localization and SLAM, our system does not allow for feature matching on RGB level, e.g. using SIFT features. Also, when dealing with comparably simple geometries, that is objects like cups, blocks or dishes vs. room scenes, getting stuck in a local minimum is dramatically more likely. Looking at Figure 4 for example, a feasible solution might be to stitch the back of the tactile point cloud to the front of the RGB-D data, a local minimum that is difficult to detect and avoid. As we control the motion of the robot and have access to proximity data, however, there is an opportunity to search for the object starting from the viewpoint of the camera. For example, we could approach the staircase from its visible side and stop the motion shortly before contact is made, preventing the hand from unintentionally bumping the object. After using the finger sensors to scan the object through light touch, the resulting data points could then be used as constraints in a RANSAC algorithm and less rely on noisy feature descriptors for better

[1] www.roboticmaterials.com

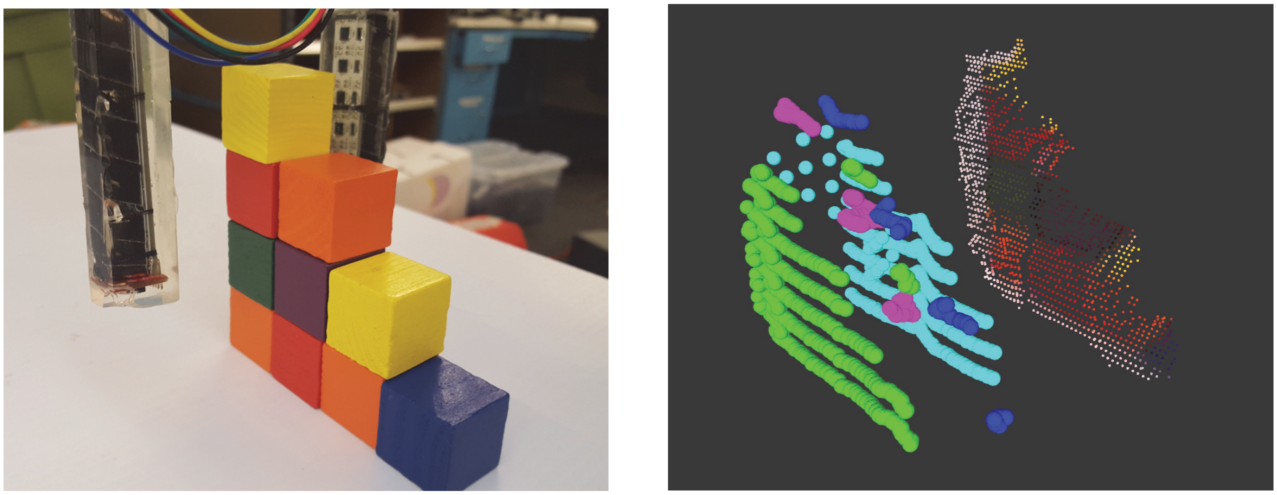

Figure 4: Left: Scanning a staircase of blocks using the finger sensors on the custom designed Baxter gripper. Right: 3D point cloud from the array of finger sensors compared with 3D point cloud from Asus Xtion depth sensor. The finger sensors provide precise information on the object and can possibly be used to gather more information of the surfaces of objects not seen by the depth sensor alone.

calibration.

We expect to show how we can use contact sensing as a feedback loop to better calibrate the camera, fusing the point clouds shown in Figure 4. With better calibration, perception and object location will be more accurate, raising the success of grasping and manipulation. Lastly, we hope to show that with improved calibration for both the camera and the sensors, we can merge the crude point cloud from proximity measurements shown in Figure 2 to detect hidden features and further improve the success of grasping and manipulation.

References

Calli, B.; Walsman, A.; Singh, A.; Srinivasa, S.; Abbeel, P.; and Dollar, A. M. 2015. Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols. *arXiv preprint arXiv:1502.03143*.

Dang, H., and Allen, P. K. 2014. Stable grasping under pose uncertainty using tactile feedback. *Autonomous Robots* 36(4):309–330.

Hsiao, K.; Kaelbling, L. P.; and Lozano-Pérez, T. 2011. Robust grasping under object pose uncertainty. *Autonomous Robots* 31(2-3):253–268.

Ma, L.; Ghafarianzadeh, M.; Coleman, D.; Correll, N.; and Sibley, G. 2015. Simultaneous localization, mapping, and manipulation for unsupervised object discovery. In *IEEE International Conference on Robotics and Automation*, 1344–1351.

Patel, R.; Alastuey, J. C.; and Correll, N. 2016. Improving grasp performance using inhand proximity and dynamic tactile sensing. In *Int. Symp. on Experimental Robotics (ISER)*.

Patel, R., and Correll, N. 2016. Integrated force and distance sensing using elastomer-embedded commodity proximity sensors. In *Proceedings of Robotics: Science and Systems*.

Patel, R.; Cox, R.; Romero, B.; and Correll, N. 2017. Improving grasp performance using in-hand proximity and contact sensing. *arXiv preprint arXiv:1701.06071*.

Saut, J.-P.; Ivaldi, S.; Sahbani, A.; and Bidaud, P. 2014. Grasping objects localized from uncertain point cloud data. *Robotics and Autonomous Systems* 62(12):1742–1754.

Tombari, F.; Salti, S.; and Di Stefano, L. 2010. Unique signatures of histograms for local surface description. In *European conference on computer vision*, 356–369. Springer.

Zhana, Q.; Liangb, Y.; and Xiaoa, Y. 2009. Color-based segmentation of point clouds. *Int Arch Photogrammetry, Remote Sens Spat Inf Sci* 38:248–252.