An Integrated System for 3D Pose Estimation in Cluttered Environments

Siddharth Raina
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sraina1@andrew.cmu.edu

Huan-Yang Chang
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
huanyanc@andrew.cmu.edu

Sambuddha Sarkar
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sambudds@andrew.cmu.edu

Man-Ning Chen
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
manning1@andrew.cmu.edu

Yiqing Cai
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
ycai1@andrew.cmu.edu

Abstract

A crucial step towards achieving complete autonomy in warehouse robots is correctly identifying objects and their pose in a cluttered environment. We present an end to end system which uses RGB images along with depth point clouds to identify and compute the 3D pose of objects in a cluttered space. We have also implemented a ground truth validation mechanism with a detailed error analysis which validates the use of our system for cluttered manipulation tasks.

1 Introduction and Problem Description

There has seen an explosive surge in the demand for warehouse automation technologies in the last decade. This surge in demand for autonomous warehouse robots has been accompanied by the need to develop autonomous warehouse pick-and-place systems which are robust enough to operate in a cluttered environment. A crucial step towards the complete autonomy of warehouse robots is a robust vision system which can use the image feed from cameras attached to the robots to accurately determine the pose of the objects which need to be manipulated. In 3, we have described our approach to design and install an end to end system for 3D pose estimation of multiple objects in a cluttered environment. Our end to end system paves the way for robot manipulation of objects in a cluttered environment with the vision of complete robot autonomy in warehouse robots.

2 Related Work

There has been work done in the problem space concerning 3D pose estimation for grippers pertaining to a preselected object in a cluttered environment. The de-facto approach is to utilixe some kind of ANN to segment the target object from the scene, employ stereo vision/depth sensing to extract 3D point cloud and then use a convergence method to estimate the pose of the end-effector of the robot to successfully clasp the target.
Sumi, Yasushi, et al.[5] do a 3D object recognition which uses segment-based stereo vision. An object is identified in a cluttered environment and its position and orientation (6 dof) are determined accurately enabling a robot to pick up the object and manipulate it. The object can be of any shape (planar figures, polyhedra, free-form objects) and partially occluded by other objects. Segment-based

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

stereo vision is employed for 3D sensing.

Comparatively Shin, Yong-Deuk, et al.[6] use a grasping strategy that is composed of the approaching vector, opposition vector, and grasping type. In this paper, they use the iterative closest point (ICP) algorithm for recognizing and estimating the pose of an object.

3 Model

3.1 The Pipeline

The functional architecture of our model is described in Figure 1. Each component of the pipeline is explained in detail in the further sections.

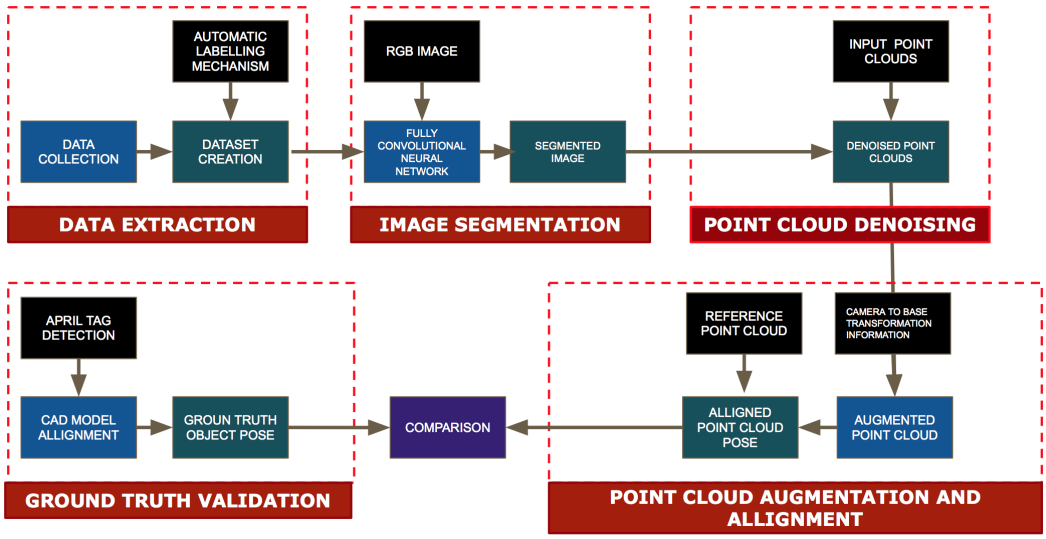


Figure 1: Architecture of our Pipeline

3.2 Data Extraction

We used an ABB Robotic Arm (2) to capture data from the Realsense camera 3. We captured images of objects from the YCB dataset. The YCB dataset has a variety of common household objects and is a famous dataset used in robotic manipulation tasks as a reliable benchmark. The data obtained from this step includes: the robot pose, RGB and depth images. Furthermore, since the transformation from RGB camera to depth camera is known, we can also get the images aligned to RGB or depth frames. This can be seen from 4.

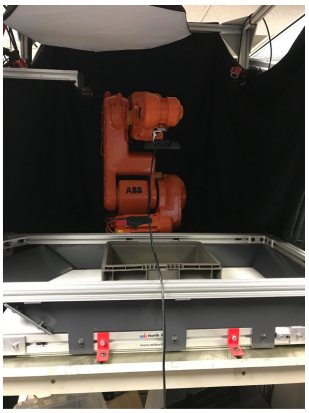


Figure 2: ABB Robotic Arm used for Data Extraction

Figure 3: RealSense SR300 camera

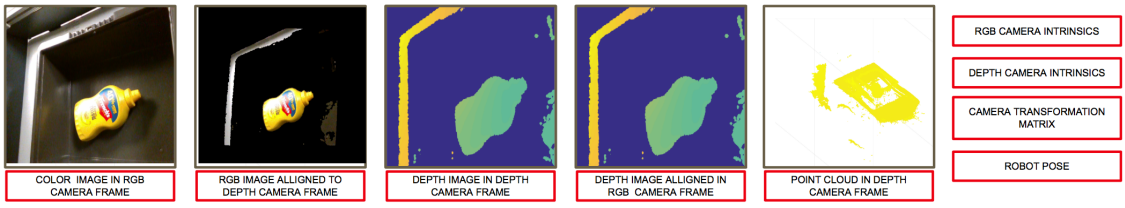


Figure 4: Data Extraction

3.2.1 Automatic Labeling Mechanism

To make the project more general and can be expanded in the future, automatic labelling is imported into our project. Objects are substracted and labeled from depth images and fed to the next step, fully convolutional network, for segmentation. This can be illustrated from 5.

Figure 5: The Automatic Labeling Mechanism

3.2.2 Dataset Generation

3.3 Object Detection and Image Segmentation

In order to get de-noised point cloud, first we need to get the segmented region of the object on the image plane and to identify the class of the object. To achieve this goal, we implemented fully convolutional network to pixel-wisely assigned each pixel to one of the 14 classes (13 objects + background) . We used 2000 images and its corresponding labeled data to train the network for 300,000 iterations, fine-tuning was conducted based on voc8s pre-trained model. The final mean accuracy per class is 0.91, and the mean Intersection of Union per class is 0.86. The object detection and segmentation pipeline is shown in 6. The sample segmentation result is shown in 7. Post-processing is also needed to remove the noise and fill in small holes in the segmented region to get better results.

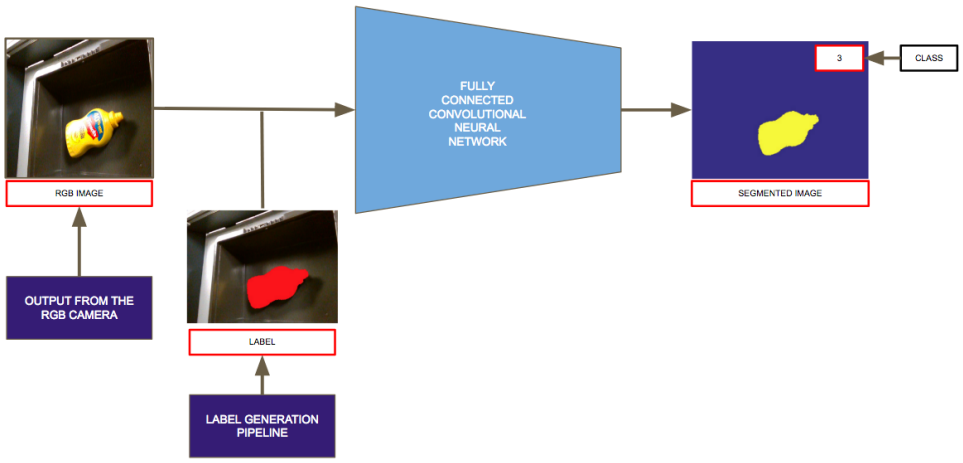


Figure 6: Object Segmentation and class identification

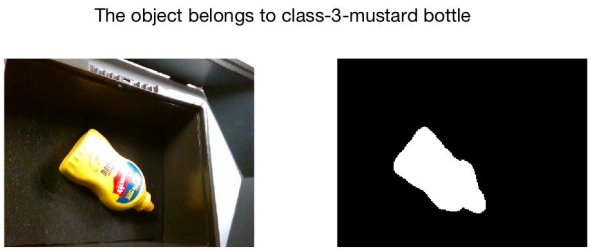


Figure 7: Sample segmentation result

3.4 Point Cloud De-noising

The Point Cloud De-noising mechanism can be illustrated from 8. The stages of the de-noising process can be briefly described as:

1. Transforming the point cloud from depth frame to RGB frame.
2. Using the Camera intrinsic and extrinsic parameters to project the points of the point cloud in the RGB camera frame.
3. De-noising the point cloud image using the segmented image from 3.3.
4. Extracting the De-noised points and projecting them back into the point cloud space.

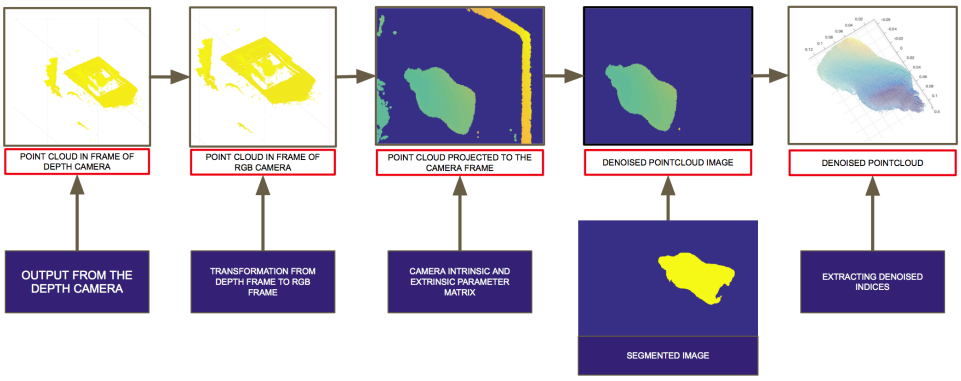
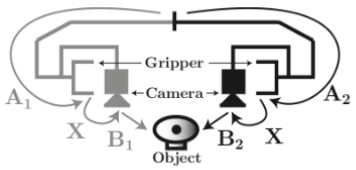


Figure 8: Point Cloud De-noising

3.5 Transformation Matrix finding

In order to find the transformation Matrix from camera to Robot base(T_C^B),We needed the transformation matrix from camera. However, this matrix was not easily to get, because it depended on how we mount the camera and how manufacture design the camera itself. Instead of directly calculate these transformation matrices, we used the least square optimization method to find out the transformation matrix. Basing on the framework we found in the paper, we need two transformation matrices, one is transformation matrix from Robot Base to End of Effector and transformation matrix from Camera frame to Object frame. The information we had in hand was the transformation matrix from Robot Base frame to end of effector frame given by ABB robot arm itself.And then we used April tag to get the transformation matrix from camera frame to April tag. By gathering this two information in different robot pose, we could use the least square optimization to get the Transformation from End of Effector to Camera frame(T_E^C).



$$A_1XB_1 = A_2XB_2 \Leftrightarrow A_2^{-1}A_1X = XB_2B^{-1} \qquad AX = XB$$

A_i: Transformation matrix from Robot Base frame to Camera Frame
B_i: Transformation matrix from Camera Frame to Object Frame(April tag)
X: Transformation matrix from End of Effector frame to Camera Frame

Figure 9: optimization from transformation matrix

3.6 Point Cloud Registration

In the point cloud registration, We used the iterative closest point(ICP) to align two point cloud and get the transformation matrix from Object to Camera frame. ICP is an algorithm employed to minimize the difference between two clouds of points. However, The ICP method was not very robust in our case and finding a good initialization is very important for a good alignment result. We initialize the position by register the center of the reference mesh and the center of our de-noised dense point cloud, and we initialize the orientation by fitting the de-noised point cloud to a cylinder and then generate the reference rotation vector. Based on the initialization, we then conduct ICP to minimize the root mean square error and generate the final 3D pose estimation of the object.

3.7 Ground Truth Validation Mechanism

By knowing the size of April tags and the camera intrinsic parameters, we could get the Translation matrix from April tag frame to the camera frame(T_A^C). Then by visually finding the translation matrix from April tag frame to object frame(T_O^A). By knowing these two information and the prior translation matrix(T_B^C), we could get the translation matrix from object to Robot base frame. On the other hand, We could also get Translation matrix from Object frame to Robot base frame from our proposed method.

$$T_O^B = T_E^B * T_C^E * T_A^C * T_O^A \qquad T_O^B = T_E^B * T_C^E * T_D^C * T_O^D$$

$$from\ April\ Tag \qquad from\ our\ proposed\ method$$

In order to evaluate the pose, we had to figure out a method to find the ground truth of the object pose. April Tag is a visual fiducial system, useful for a wide variety of tasks including augmented reality, robotics, and camera calibration. In out implementation, we attached April tag on the target object, and then found out the relationship between the center and April tag(Figure 10).

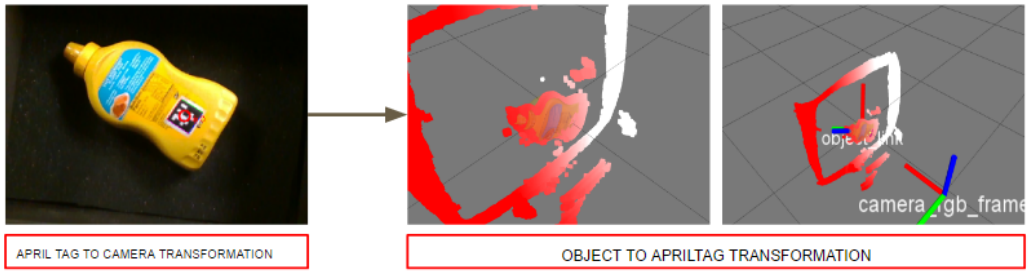


Figure 10: Using April tags to get ground truth

By these two translation matrices, we could compare the proposed method by evaluation the distance of the object's center in the robot frame by these two transforms and the difference in angle. The limitation of using April tag was it only work when the camera could clearly see the April tag and the view angle could also influence the accuracy of the detection of April tag. Because of that, we could only put the object in some position.

4 Results

In the final evaluation, we chose three different objects from YCB dataset, which were a mustard bottle, a sugar box, and a banana. The reason why we chose these three was because each of them stood as a specific shape of the object in the YCB dataset which could good enough to evaluate our proposed method. The mustard and the sugar box represent the object that was quite regular that could be simply taken as a cylinder or a box. On the other hand the banana was quite different, then we could take it as an arbitrary shape.
We expressed the evaluation result as six numbers which mean the distance between the center and the angle between the each axis. In the average result, for the mustard, the distance of center were less than 0.5cm in x and y axis and less than 3cm in z axis; the angle error is around 15 degree. For the sugar box, the distance of center were less than 0.1cm in y-axis and less than 2cm in x and z-axis; the angle error is around 13 degree. For the banana, the distance of center were less than 0.1cm in y -axis and less than 1.5cm in x and z-axis; the angle error is around 30 degree.

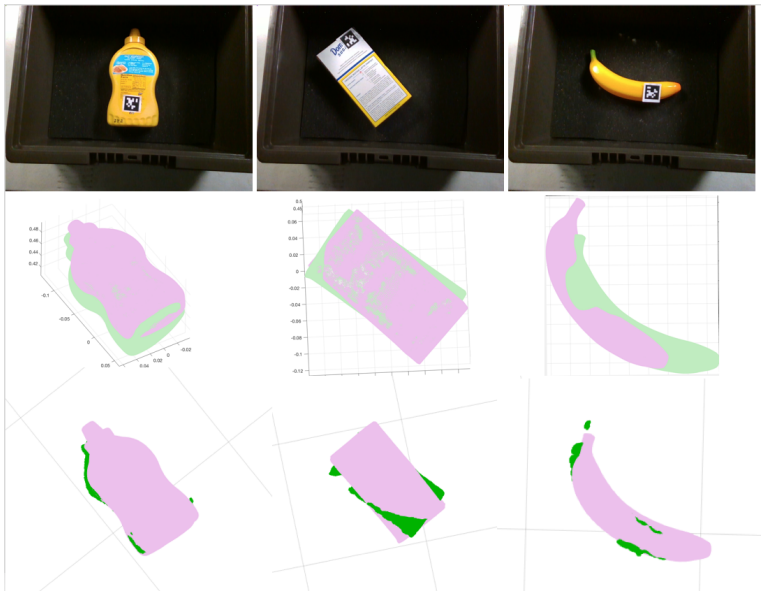


Figure 11: Pose estimation compared with ground truth(purple); first row: RGB images, second row: Pose comparison, third row: point cloud alignment

	x	y	z	x_r	y_r	z_r
mustard_1	0.0063	0.0003	0.0452	9.0073	12.6790	14.8256
mustard_2	0.0047	0.0131	0.0119	26.3415	23.3784	14.2482
mustard_3	-0.0063	0.0050	0.0305	8.4059	22.0455	20.5965
mustard_4	0.0048	0.0043	0.0182	19.4923	19.1843	3.4758
mustard_5	-0.0012	-0.0001	0.0234	18.4200	19.4468	6.2931
average	0.00166	0.00452	0.02584	16.3334	19.3468	11.88784
stdev	0.0052955 64	0.0053162 02	0.0127981 64	7.5995330 75	4.1240981 84	6.9301158 86

Figure 12: Error on pose estimation result compared with ground truth on five different pose of mustard bottle(including positional(meter) and rotational(degree) error along each axis in 3D space)

	x	y	z	x_r	y_r	z_r
sugar_1	-0.0024	-0.0101	0.0161	8.6738	2.4549	8.8745
sugar_2	-0.0073	-0.0011	0.0488	2.5708	4.7728	5.4078
sugar_3	0.0023	-0.0097	0.0170	11.9969	5.0724	2.5388
sugar_4	0.0377	0.0087	0.0112	21.2249	21.2296	5.8643
sugar_5	0.0530	-0.0052	-0.0035	23.8823	31.5728	21.6079
average	0.01666	-0.00348	0.01792	13.66974	13.0205	8.85866
stdev	0.0269576 15	0.0077377	0.0191161 45	8.8363044 26	12.790679 32	7.4725937 34

Figure 13: Error on pose estimation result compared with ground truth on five different pose of sugar box(including positional(meter) and rotational(degree) error along each axis in 3D space)

	x	y	z	x_r	y_r	z_r
banana_1	0.0100	-0.0065	0.0152	27.0777	28.5586	14.1921
banana_2	-0.0101	-0.0089	0.0093	49.3793	30.1653	38.1685
banana_3	-0.0112	0.0098	0.0222	50.6751	32.0744	40.6614
banana_4	-0.0288	-0.0017	0.0107	21.0828	41.0284	37.2125
banana_5	-0.0109	0.0123	0.0141	10.3100	15.5421	17.5230
average	-0.0102	0.001	0.0143	31.70498	29.47376	29.5515
stdev	0.0137431 8	0.0095744 45	0.0050304 08	17.778038 68	9.1607850 33	12.619127 04

Figure 14: Error on pose estimation result compared with ground truth on five different pose of banana(including positional(meter) and rotational(degree) error along each axis in 3D space)

5 Work division

The work division is briefly summarized in Table 1

Table 1: Work Division

Task	Division
Data Extraction	Huan-Yang Chang, Yiqing Cai, Siddharth Raina, Man-Ning Chen, Sambuddha Sarkar
Automatic Labeling Mechanism	Man-Ning Chen, Siddharth Raina, Sambuddha Sarkar
Object Detection and Segmentation	Yiqing Cai, Man-Ning Chen
Point Cloud De-noising	Siddharth Raina, Sambuddha Sarkar
Transformation Matrix ($T_{camera}^{end-effector}$) Modeling	Huan-Yang Chang
Point Cloud Registration	Huan-Yang Chang, Yiqing Cai, Man-Ning Chen, Siddharth Raina
Ground Truth Validation Mechanism	Huan-Yang Chang

6 References

[1] Sumi, Yasushi, et al. "3D object recognition in cluttered environments by segment-based stereo vision." International Journal of Computer Vision 46.1 (2002): 5-23.

[2] Shin, Yong-Deuk, et al. "Integration of recognition and planning for robot hand grasping." Ubiquitous Robots and Ambient Intelligence (URAI), 2013 10th International Conference on. IEEE, 2013.

[3] Choi, Changhyun, and Henrik I. Christensen. "3D pose estimation of daily objects using an RGB-D camera." Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.