SegICP: Integrated deep semantic segmentation and pose estimation

SegICP: Integrated Deep Semantic Segmentation and Pose Estimation

Jay M. Wong, Vincent Kee[†], Tiffany Le[†], Syler Wagner, Gian-Luca Mariottini,
Abraham Schneider, Lei Hamilton, Rahul Chipalkatty, Mitchell Hebert, and David M.S. Johnson
Draper, Cambridge, MA, USA

Jimmy Wu, Bolei Zhou, and Antonio Torralba
Massachusetts Institute of Technology, Cambridge, MA, USA

***Abstract*— Recent robotic manipulation competitions have highlighted that sophisticated robots still struggle to achieve fast and reliable perception of task-relevant objects in complex, realistic scenarios. To improve these systems' perceptive speed and robustness, we present SegICP, a novel integrated solution to object recognition and pose estimation. SegICP couples convolutional neural networks and multi-hypothesis point cloud registration to achieve both robust pixel-wise semantic segmentation as well as accurate and real-time 6-DOF pose estimation for relevant objects.Our architecture achieves 1 cm position error and $<$ 5° angle error in real time *without* an initial seed. We evaluate and benchmark SegICP against an annotated dataset generated by motion capture.**

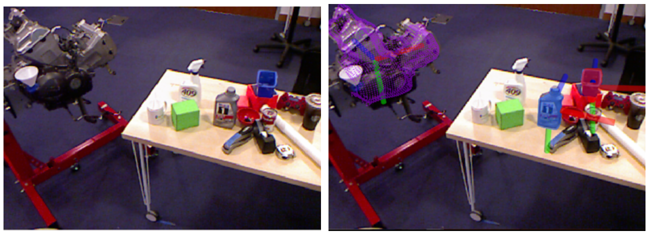


Fig. 1: Given an RGB image (left) and depth frame, our SegICP approach segments the objects in a pixel-wise fashion and estimates the 6 DOF pose of each object with 1 cm position and 5° angle error (right).

I. Introduction

To achieve robust, autonomous operation in unstructured environments, robots must be able to identify relevant objects and features in their surroundings, recognize the context of the situation, and plan their motions and interactions accordingly. Recent efforts in autonomous manipulation challenges such as the DARPA Robotics Challenge [1] and the Amazon Picking Challenge [2] resulted in state-of-the-art perception capabilities enabling systems to perceive, reason about, and manipulate their surroundings. However, existing object identification and pose estimation solutions for closed-loop manipulation tasks are generally (1) not robust in cluttered environments with partial occlusions, (2) not able to operate in real-time (<1 Hz), (3) not sufficiently accurate [3], or (4) incapable of high accuracy without good initial seeds [4].

We present a novel perception pipeline that tightly integrates deep semantic segmentation and model-based object pose estimation, achieving real-time pose estimates with a median pose error of 1 cm and $<5°$. Our solution (referred to as *SegICP*) uses RGB-D sensors (and proprioceptive information when available) to provide semantic segmentation of all relevant objects in the scene along with their respective poses (see Figure 1) in a *highly parallelized* architecture.

The main contributions of this manuscript are as follows:

1) A *highly parallelized* approach to integrated semantic segmentation and multi-hypothesis object pose estimation with 1 cm accuracy with a single view operating at 70–270 ms (4–14 Hz) *without any prior pose seeds*.
2) A novel metric to score the quality of point cloud registration, allowing for autonomous and accurate pose initialization over many potential hypotheses.
3) An efficient automatic data-collection framework for acquiring annotated semantic segmentation and pose data by using a motion capture system.
4) Analysis and benchmarking of our SegICP pipeline against the automatically annotated object poses.

II. Related Work

Our approach builds on the substantial literature devoted to robot perception of mobile manipulation task environments and the relevant objects therein. Robot systems must be able to first identify entities that pertain to the task at hand and reason about their relative poses to eventually manipulate and interact with them. Accordingly, we discuss the relevant literature in object recognition and pose estimation.

Object Recognition. Semantic segmentation, which assigns each pixel in an image to one of a set of predefined categories, effectively solves the object recognition problem. This approach is in contrast to that of many object recognition systems, which only output bounding boxes around objects of interest [5–7]. Although the winning team for the 2015 Amazon Picking Challenge used a bag-of-words approach [8] instead of per-pixel categorization, there are multiple advantages to retaining the spatial position of every object. Particularly in robotic applications such as grasping or autonomous driving, semantic segmentation enables a higher resolution representation of the arrangement, identities of objects in a cluttered scene, and effectively addresses

Draper Fellows[†] associated with both Draper and MIT. Any opinions, findings, conclusions, or recommendations expressed in this material are solely those of the author(s) and do not necessarily reflect the views of these organizations. We thank Leslie Pack Kaelbling, Tomás Lozano-Pérez, Scott Kuindersma, and Russ Tedrake for their insight and feedback.
Corresponding author: Jay M. Wong jmwong@draper.com

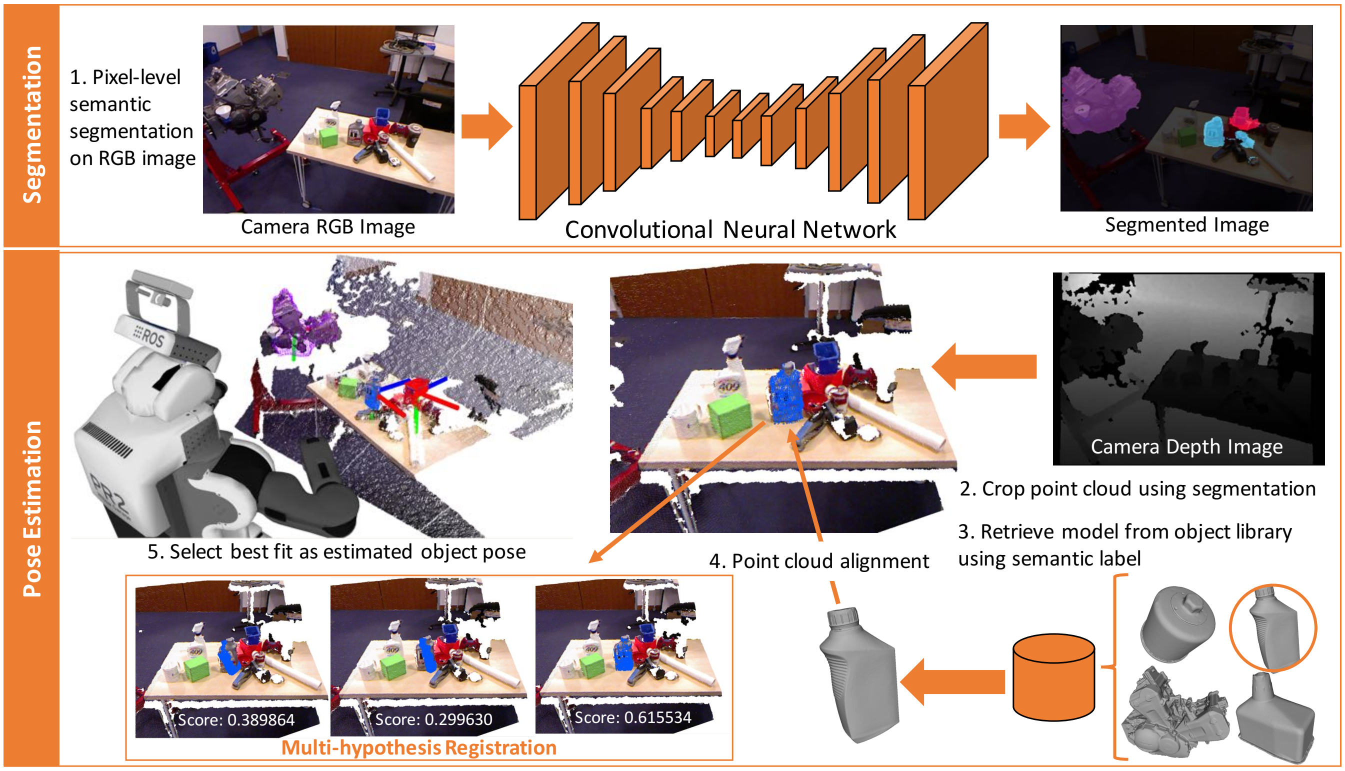

Segmentation
1. Pixel-level semantic segmentation on RGB image
Camera RGB Image
Convolutional Neural Network
Segmented Image
Pose Estimation
Camera Depth Image
2. Crop point cloud using segmentation
3. Retrieve model from object library using semantic label
4. Point cloud alignment
5. Select best fit as estimated object pose
Score: 0.389864
Score: 0.299630
Score: 0.615534
Multi-hypothesis Registration

Fig. 2: The full SegICP pipeline operating in a cluttered environment. The system detects objects relevant to the automotive oil change task and estimates a 6-DOF pose for each object. The colored overlay pixels in the segmented image (top right) correspond to a blue funnel (red), an oil bottle (blue), and the engine (purple), as detected by the Kinect1 mounted on top of a PR2 robot. Selected multi-hypothesis registrations for the oil bottle object (bottom left) are shown with their respective alignment scores. The hypothesis registrations are evaluated in parallel to determine the optimal object pose.

self-occlusions. Previous approaches to object recognition have used classifiers with hand-engineered features [9, 10] to generate either bounding-box object locations or noisy pixel-wise class predictions, which are then smoothed using CRFs [11]. Recent work in computer vision has shown that convolutional neural networks (CNNs) considerably improve image classification [12, 13]; CNNs originally trained for image classification can be successfully re-purposed for dense pixel-wise semantic segmentation [14–18]. Such approaches generally retain the lower level feature detectors of image classification models such as AlexNet [13] or VGG [19] and stack on additional layers, such as convolution [14], deconvolution [15], or dilated convolution [18].

Pose Estimation. While semantic segmentation is able to identify and locate objects in 2D images, pose estimation refines object location by also estimating the most likely 6-DOF pose of each identified object. Previous approaches to this task have used template matching, which can recover the pose of highly-textured objects [20] [21] using local features such as SIFT [22]. For RGB-D images, use of stable gradient and normal features has been demonstrated with LINEMOD [23, 24]. Approaches using parts-based models have also been successful [25–27]. However, these methods are not robust to variations in illumination or to scene clutter [3]. While a class of point cloud registration algorithms attempt to solve the global optimization problem [6], such approaches rely on surface normals features and degrade when objects are generally flat, have low quantities of informative features, or exhibit potentially ambiguous geometries.

A widely accepted approach to pose estimation is the class of iterative closest point (ICP) registration algorithms [28–30]. These approaches usually require initialization close to the global optima as gradient-based methods tend to fall into poor local minima and are not robust to partial or full occlusions [4]. Most relevant to our work, Team MIT-Princeton demonstrated promising results in the Amazon Picking Challenge using multiple views with a fully convolutional neural network to segment images and fit 3D object models to the segmented point cloud [3]. However, their pose estimation system was slow (∼1 s per object) and showed high position and angle errors (5 cm and ∼15°). We advance this prior work by presenting a novel metric for scoring model registration quality, allowing accurate initial pose estimation through multi-hypothesis registration. Further, we emphasize an order of magnitude speedup by leveraging a *highly parallelized* design that operates over all objects simultaneously. We couple these advances with an efficient data collection pipeline that automatically annotates semantic segmentation labels and poses for relevant objects.

III. Technical Approach

We present SegICP, a novel perceptual architecture that handles sensor input in the form of RGB-D and provides a semantic label for each object in the scene along with its associated pose relative to the sensor. SegICP acquires and tracks the 6-DOF pose of each detected object, operating at ~70 ms per frame (270 ms during initialization phase) with 1 cm position error and $< 5^\circ$ angle error, and can robustly deal with prolonged occlusions and potential outliers in the segmentation with a Kalman filter. SegICP achieves this using an object library approach to perception, referencing scanned 3D models of known objects, and performs 3D point cloud matching against cropped versions of these mesh models. In our architecture, as outlined in Figure 2, RGB frames are first passed through a CNN which outputs a segmented mask with pixel-wise semantic object labels. This mask is then used to crop the corresponding point cloud, generating individual point clouds for each detected object. ICP is used to register each object's point cloud with its full point cloud database model and estimate the pose of the object with respect to the sensor.

A. Semantic Segmentation by Neural Networks

Contrary to classical segmentation problems, we are specifically concerned with generating *appropriate* masks over the depth map to aid accurate pose estimation. In an attempt to address this, we experimented with various CNN architectures that semantically segment known objects of interest. We explored two different CNN architectures, SegNet [14] and DilatedNet [18] (further discussed and elaborated in Section IV-A). Of the two networks, we found that the best model for our SegICP pipeline was SegNet, a 27-layer, fully convolutional neural network with 30 million parameters. The network was trained on cropped and downsampled images from the training set (to 320×320 pixels) consisting of eight object classes (including background) using the cross entropy criterion coupled with data augmentation consisting of image rotations, crops, horizontal and vertical flips, and color and position jitter. We further elaborate on the acquisition of annotated training data in Section III-C.

B. Multi-Hypothesis Object Pose Estimation

The resulting segmentation is used to extract each object's 3D point cloud from the scene cloud. The identity of each segmented object (the object's semantic label) predicted by SegNet is then used to retrieve its corresponding 3D mesh model from the object model library. The mesh model is converted into a point cloud representation, downsampled, and registered against its respective segmented point cloud.

Point cloud registration is divided into two phases: *acquisition* and *tracking*. The objective of the acquisition phase is to find the initial optimal alignment between each object's model and its corresponding scene point cloud. This alignment is used to determine the visible side of the model (*model crop*) and to initialize the tracking phase, whose objective is to fuse camera and robot motion information to maintain an accurate, real-time pose estimate of the object even during camera motion and potential occlusions. We use a point-to-point ICP [31] algorithm for registration. A contribution of this paper is the model-to-scene alignment metric that is used to determine the registration quality as well as switching between acquisition and tracking phases.

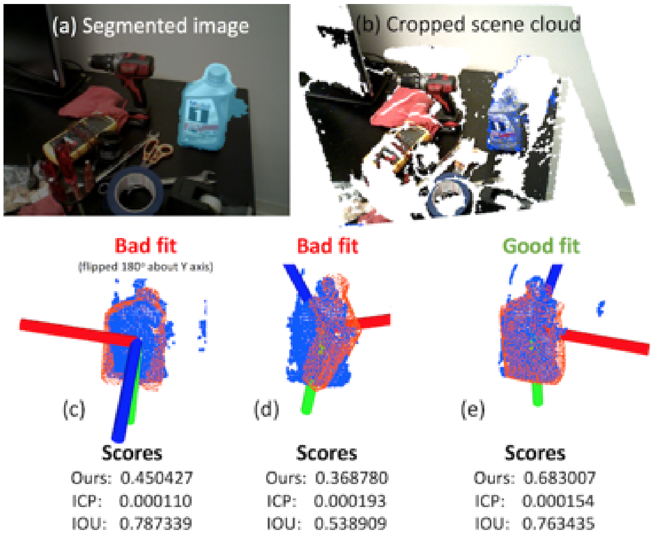


Fig. 3: Examples of Multi-Hypothesis Registration Ranking: The segmentation generated in (a) is used to produce the cropped scene cloud highlighted in blue (b). Panels (c-e) illustrates the registration of various candidate model crops (orange) to the cropped scene cloud (blue), along with their respective alignment scores.

The Acquisition Phase. The acquisition phase finds the initial optimal alignment and crop of the object's mesh model with the current point cloud. Multiple candidate crops are obtained by rendering the visible object's model at various azimuth and elevation angles and cropping the model to keep only the front face. Each of the candidate crops is initialized at the median position of the object's scene point cloud in order to remove segmentation outliers and prevent ICP from converging to incorrect local minima. In parallel, each candidate crop is run through a few iterations of the tracking phase to achieve a pose hypothesis.

A novel model-to-scene alignment metric is evaluated on each candidate model crop. The motivation behind the metric is to determine whether a candidate cloud can align well with the object cloud by finding the number of points in the candidate cloud with a unique corresponding match in the object's cloud. Letting $\mathcal{M}_i$ be the set of points in the candidate crop point cloud and S be the set of points in the segmented object scene cloud, the alignment metric is given by: $\text{a}(\mathcal{M}_i, S) = \frac{|c|}{|\mathcal{M}_i|}$ where c is the set of points in $\mathcal{M}_i$ with unique corresponding points in S at most τ meters away. To compute the metric, SegICP first builds a kd-tree with S and perform radius searches with a radius of τ meters from every point in $\mathcal{M}_i$. Each point in $\mathcal{M}_i$ is mapped to the closest point in S within τ that has not been already mapped to another point in $\mathcal{M}_i$, and then is added to c.

Figure 3 show examples of model crops and their respective scores. In particular, we illustrate metrics such as the ICP fitness score (a Euclidean error score) and intersection over union (IOU)[1] do not effectively distinguish good registrations from erroneous ones. In comparison, our proposed metric

[1] IOU is computed between the predicted segmentation and the projection of the registered model into the camera optical frame.

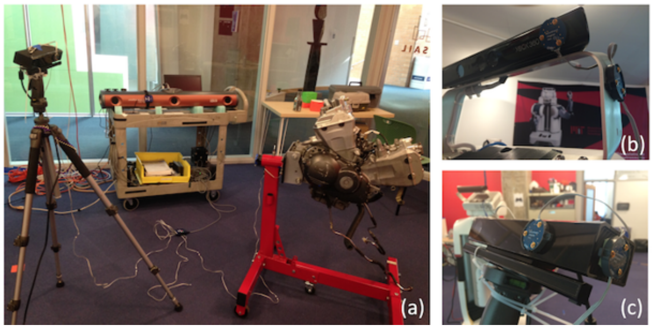


Fig. 4: Motion Capture System: Setup using the NDI 3D Investigator Motion Capture System (a). We mount small, circular active markers on the RGB-D camera and the objects for pose measurement. Examples of these markers are shown on the PR2's Kinect1 (b) and on the Kinect2 (c)

addresses these immediate shortcomings present on objects with high degrees of symmetry (e.g. the oil bottle). If any candidate scores are above a threshold ϵ, SegICP switches to the tracking phase for future frames.

The Tracking Phase. The candidate model pose and crop with the highest alignment score are used to initialize the tracking phase. In order to make the tracking procedure robust to imperfections on the boundary of the object's segmentation, the object's scene point cloud is further pruned by removing points outside a bounding box of the latest registered model pose. The pose obtained by registration is used as a measurement update in a Kalman filter to track each object's 6-DoF pose and velocities. By fusing known camera motions from the available odometry of the robot, the filter is able to handle temporary object occlusions and outlier pose estimates. Our alignment metric is evaluated on the fit to measure the uncertainty of the current pose measurement and to inform the Kalman filter accordingly. If the score goes below a minimum threshold θ, the Kalman filter propagates the objects' pose based on odometry (and until a maximum pose uncertainty) while switching back to acquisition mode.

C. Automatically Annotating Training Data

We trained SegNet on 7500 labeled images of indoor scenes consisting of automotive entities (e.g. engines, oil bottles, funnels, etc). Of these images, about two-thirds were hand-labeled by humans (using LabelMe [32]) while the remaining third was generated automatically by a 3D Investigator$^{\text{TM}}$ Motion Capture (MoCap) System and active markers placed on our cameras and objects (shown in Figure 4). The training images span multiple sensor hardware (Microsoft Kinect1, Asus Xtion Pro Live, Microsoft Kinect2, and Carnegie Robotics Multisense SL) each with varying resolutions (respectively, 640×480, 640×480, 1280×1024, and 960×540). However, obtaining large datasets for segmentation and pose is difficult. As a result, we present a motion capture system to automatically annotate images shown in Figure 5. Active markers are placed on the engine stand and on the corner of the table. Known transformations via MoCap are then used to segment the image by projecting a scanned object mesh using the transform into the camera optical frame, thus generating annotated segmentation and object pose data.

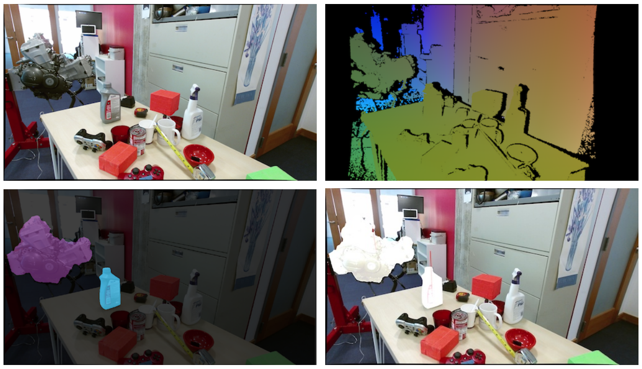


Fig. 5: Automatic Motion Capture Annotation: Given input RGB and encoded depth images (top row), the automatic labeling system outputs the segmentation and object poses in axis angle format (bottom row).

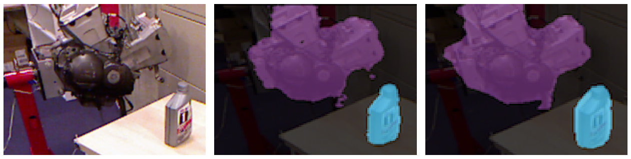


Fig. 6: SegNet and DilatedNet: outputs respectively (middle, right) given the same input RGB input image (left) from the PR2's Kinect1; SegNet appears to generate tighter segmentation compared to DilatedNet.

IV. Evaluation

We benchmark SegICP on a dataset consisting of 1246 annotated object poses obtained via the MoCap system.

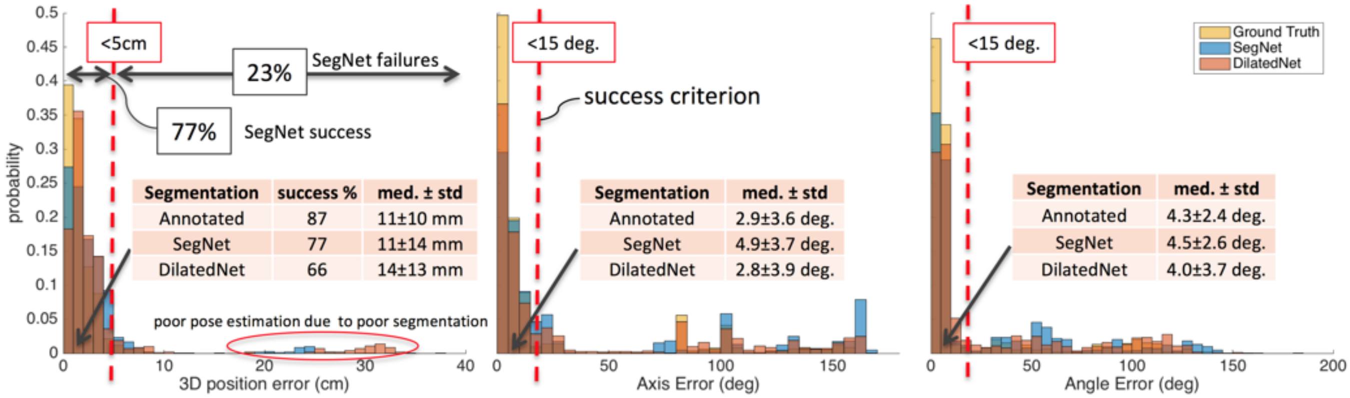
A. Semantic Segmentation Results

To categorize the influence of segmentation on pose estimation, we explored two architectures for semantic segmentation: SegNet and DilatedNet. SegNet is a computationally efficient autoencoder-decoder for pixel-wise semantic segmentation. The autoencoder architecture is essential for per-pixel classification, as it enables reconstruction of the inputs from the outputs at each layer, learning how to reconstruct the input before the final classification layer. DilatedNet makes use of dilated convolution modules to aggregate multi-scale contextual information without losing accuracy. Both network architectures adapt the convolutional layers of the VGG [19] image classification network, with SegNet using the VGG layers as its encoder and DilatedNet converting later layers into dilated convolution modules. Weights are initialized during training using a VGG-16 model pretrained on ImageNet [33]. We train both of these networks with a dataset of over 7500 annotated images (average epoch time of about an hour) and obtained the performance measures listed in Table I.

	SegNet	**DilatedNet**
IOU median (±std)	0.850 ± 0.159	0.752 ± 0.189
Precision median (±std)	0.897 ± 0.107	0.807 ± 0.183
Recall median (±std)	0.961 ± 0.164	0.965 ± 0.162

TABLE I: The performance of the semantic segmentation networks.

A key distinction between the two architectures is that DilatedNet was designed for increased recall by incorporating dilated convolution modules whereas SegNet appears to



Segmentation	success %	med. ± std
Annotated	87	11±10 mm
SegNet	77	11±14 mm
DilatedNet	66	14±13 mm

Segmentation	med. ± std
Annotated	2.9±3.6 deg.
SegNet	4.9±3.7 deg.
DilatedNet	2.8±3.9 deg.

Segmentation	med. ± std
Annotated	4.3±2.4 deg.
SegNet	4.5±2.6 deg.
DilatedNet	4.0±3.7 deg.

Fig. 7: SegICP Pose Estimation: errors between varying segmentation masks (where Annotated refers to the annotated segmentation derived from MoCap) as a result of different neural network architectures (e.g. SegNet and DilatedNet).

achieve higher precision measures. Notable visual differences are illustrated in Figure 6, where the output of SegNet and DilatedNet is displayed for the same scene. It is important to note that the *quality* of the segmentation influences the point cloud mask and has immediate impact on the performance of our point-to-pose registration pipeline for object pose estimation. Still, the following questions still remain: *Does higher segmentation IOU result in better pose? Higher precision? Higher recall?* In the next section, we perform several benchmarks to investigate these very questions.

B. *Pose Estimation Results*

The Acquisition and Tracking Phases. In our benchmarking, we used a collection of $N = 30$ model crops for each object during the acquisition phase and discovered an overall average runtime of 270 ms over a collection of thirty threads on a six-core i7-6850K. However, note that the time evaluation here is directly dependent on the number of crops and the machine's CPU. The registration of each of these crops proposed a separate object pose hypothesis (alike Figure 3), and we used a threshold of $\epsilon = 0.75$ to switch into the tracking phase, which continuously updates the object's pose using the optimal crop, operating at about 70 ms, with 45–50 ms being the neural network forward propagation (with nVidia GTX Titan X). For the kd-tree radius search to compute the metric, we used $\tau = 1$ cm.

Benchmarking. In Figure 7, we illustrate the results of evaluating SegICP on the benchmarking dataset of 1246 object pose annotations. To fully categorize the influence of the segmented mask on the final pose estimation, we ran SegICP using the annotated segmentation and the output of the two segmentation neural network architectures: SegNet and DilatedNet. These results indicate that SegNet achieves higher performance (77%) as compared to DilatedNet (66%). We categorize failure as exceeding errors of more than 5 cm in position and 15° in axis and axis angle. These failures due to segmentation errors and model crop coverage represent a class of highlighted instances in the figure. Of the successful scenes, SegICP achieves 1 cm position error and $< 5°$ angle error; this level of accuracy corresponds to about 80% of all the benchmarked instances. Further performance measures are given in Figure 7, where we show the distribution of pose estimation errors given segmentation.

Interestingly, the performance of SegICP is highly correlated with both sensor technology and calibration. When considering only the 466 Kinect1 instances (a structured light sensor with better RGB-D calibration), SegICP achieves success measures of 90%, 73%, and 72% using segmented masks from annotation, SegNet, and DilatedNet respectively; the networks appear to have comparable performance. However, when calibration is subpar, in the case of our Kinect2 (which is also a time of flight sensor), it is beneficial to bound the number of false-positive pixels (maximizing precision) to avoid acquiring misaligned points in the cropped scene cloud. From Table I, SegNet and DilatedNet achieve precision measures of 0.897 and 0.807 respectively. With the Kinect2, we observe success measures of 85%, 80%, and 62% for annotated, SegNet, and DilatedNet segmentation; the large inconsistencies with DilatedNet is as a result of poor cropped scene clouds due to excessive false-positives in the segmentation (poor precision).

Further, it appears that SegICP operates with higher performance on structured light sensors (e.g. Kinect1) compared to time of flight sensors (e.g. Kinect2). We discovered that objects with reflective surfaces (e.g. the oil bottle) with high levels of geometric symmetry and potential ambiguities result in poor ICP fits due to the deformations in the point cloud caused by time of flight. Figure 8 illustrates this particular phenomenon, where large deformities on the surface of the oil bottle is present, resulting in poor registration. Lastly, because the architecture uses a segmentation mask (generated using the RGB frames) to crop the point cloud, the sensor calibration of the RGB and depth frames is crucial for accurate pose estimation.

V. Conclusion

We present a novel, *highly parallelized* architecture for semantic segmentation and accurate pose estimation (1 cm position error and $< 5°$ angle error). Our architecture delivers immediate benefits as compared to work in the literature by not requiring an initial guess sufficiently close to the solution and by being inherently parallelizable, allowing us

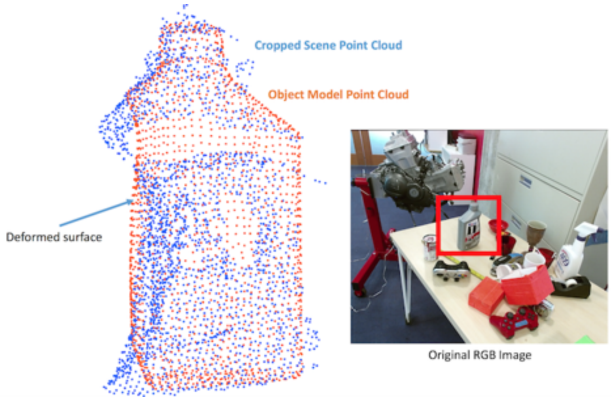


Fig. 8: Point Cloud Deformation: experienced by time of flight sensors (e.g. Microsoft Kinect2) on reflective surfaces and a culprit of object model cloud misalignments in the dataset.

to process multiple objects simultaneously in real time (70–270 ms in tracking and acquisition mode respectively). We elaborated on a motion capture approach to collecting potentially massive sets of annotated segmentation and pose data, allowing our architecture to scale rapidly to more enriched domains. Lastly, we categorized the segmentation-driven method to pose estimation by extensively investigating and benchmarking two different neural network architectures.

We are currently working to refine the perception architecture, extend the framework to incorporate much larger sets of objects and tie it with integrated task and motion planning for complex interactions in unstructured environments.

References

[1] G. Pratt and J. Manzo, "The darpa robotics challenge [competitions]," *IEEE Robotics & Automation Magazine*, vol. 20, no. 2, pp. 10–12, 2013.

[2] P. R. Wurman and J. M. Romano, "The amazonpicking challenge 2015," *IEEE Robotics and Automation Magazine*, vol. 22, no. 3, pp. 10–12, 2015.

[3] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker Jr, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge," *arXiv preprint:1609.09475*, 2016.

[4] T. Schmidt, R. A. Newcombe, and D. Fox, "Dart: Dense articulated real-time tracking." in *Robotics: Science and Systems*, 2014.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[6] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European Conference on Computer Vision*. Springer, 2016, pp. 766–782.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[8] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa, "Fast object localization and pose estimation in heavy clutter for robotic bin picking," *The International Journal of Robotics Research*, 2016.

[9] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Conference on Computer vision and pattern recognition (CVPR)*, 2008, pp. 1–8.

[10] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *European Conference on Computer Vision*. Springer, 2008, pp. 44–57.

[11] L. Ladickỳ, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr, "What, where and how many? combining object detectors and crfs," in *European conference on computer vision*. Springer, 2010, pp. 424–437.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[14] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint:1505.07293*, 2015.

[15] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[17] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *arXiv preprint arXiv:1608.05442*, 2016.

[18] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint:1511.07122*, 2015.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint:1409.1556*, 2014.

[20] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[21] D. G. Lowe, "Local feature view clustering for 3d object recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[22] ——, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[23] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian conference on computer vision*. Springer, 2012, pp. 548–562.

[24] R. Rios-Cabrera and T. Tuytelaars, "Discriminatively trained templates for 3d object detection: A real time scalable approach," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2048–2055.

[25] S. Savarese and L. Fei-Fei, "3d generic object categorization, localization and pose estimation," in *11th International Conference on Computer Vision*, 2007, pp. 1–8.

[26] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Teaching 3d geometry to deformable part models," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3362–3369.

[27] J. J. Lim, A. Khosla, and A. Torralba, "Fpm: Fine pose parts-based model with 3d cad models," in *European Conference on Computer Vision*. Springer, 2014, pp. 478–493.

[28] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.

[29] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *International Conference on 3-D Digital Imaging and Modeling*. IEEE, 2001, pp. 145–152.

[30] C. Yuan, X. Yu, and Z. Luo, "3d point cloud matching based on principal component analysis and iterative closest point algorithm," in *International Conference on Audio, Language and Image Processing (ICALIP)*. IEEE, 2016, pp. 404–408.

[31] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[32] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1, pp. 157–173, 2008.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.