Bootstrapping Object Pose Detection

Tushar Agrawal
SCS, RI, MRSD
tagrawa1@andrew.cmu.edu

Cole Gulino
SCS, RI, MRSD
cgulino@andrew.cmu.edu

Erik Sjoberg
SCS, RI, MRSD
esjoberg@andrew.cmu.edu

1 Introduction

Object pose detection has many applications in the realm of robotics including industrial applications for robots that pick and place objects. Many state of the art algorithms are designed to use computer vision and other perception techniques (iterative closest point, etc.) in order to estimate the pose of an object on the table. Often dense 3D models of the objects and classic computer vision techniques are utilized in realms where the object's shape is readily known to improve perception accuracy.

During our literature review, we identified three promising approaches to using deep learning to perform 6 degree-of-freedom (DOF) pose estimation of known objects using an RGB-D sensor. The first of these approaches described in [1] learns multi-dimensional descriptors which encode the appearance of an entire object from a given angle, then at test-time uses K-NN to find the nearest matching descriptor. The second approach described in [3] builds on [1] by learning descriptors for multiple local patches of an object which have been scaled to a standard physical size, then using a filtering scheme to accumulate votes of likely object positions. A third method found in [4] dispenses with the descriptor approach, and directly regresses on the pose (represented as a quaternion) using a siamese architecture [5]. These three architectures are discussed in more detail below.

In the first paper [1] described in Figure 1, multi-dimensional descriptors are learned which encode the appearance of an entire object from a given angle. Later, at test-time, K-NN is used to find the nearest matching descriptor which should correspond to the appropriate view of the corresponding object. A major benefit of this approach is that, with an appropriate loss function, it can be trained to place descriptors for different objects in distinct locations in the multidimensional descriptor space, allowing for both the object identity and pose to be extracted in a single step.

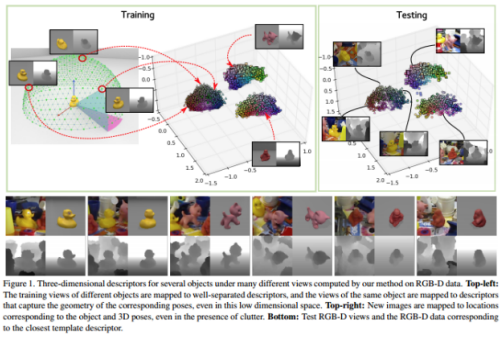

Figure 1. Three-dimensional descriptors for several objects under many different views computed by our method on RGB-D data. **Top-left:** The training views of different objects are mapped to well-separated descriptors, and the views of the same object are mapped to descriptors that capture the geometry of the corresponding poses, even in this low dimensional space. **Top-right:** New images are mapped to locations corresponding to the object and 3D poses, even in the presence of clutter. **Bottom:** Test RGB-D views and the RGB-D data corresponding to the closest template descriptor.

Figure 1: Method Described in the Wohlhart Paper

The next paper that we looked at [3], described in Figure 2, builds on [1] by learning descriptors for multiple local patches of an object using an auto-encoder, with all of these patches scaled to a standard physical size using the depth information from the RGB-D camera. A code-book is created from the set of known object patch / object center labels. A dense set of patches are sampled from the test image, and these patches are passed through the trained auto-encoder, giving potential matches

10807 Class Project (2016).

amongst the various known object patches. A filtering scheme is then used to accumulate votes (with confidence values above a threshold) of the most likely object identities and positions, resulting in a multi-modal set of labeled 6-dof object centers.

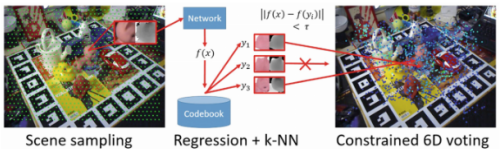

Figure 2: Method Described in the Kehl Paper

The final paper that we looked at [4], illustrated in Figure 3, a siamese architecture as described in [5] is used to learn a similarity function which enforces a distance relationship between the feature space of the image and the pose of the object. The key to their improved performance is the the additional loss term which pushes the feature-space representation to map more directly to the pose-space, resulting in the network learning more discriminative features. At test time, the pose in angle space is a direct output of the network, in contrast with the first two approaches which must perform lookups in a codebook using K-NN or similar techniques.

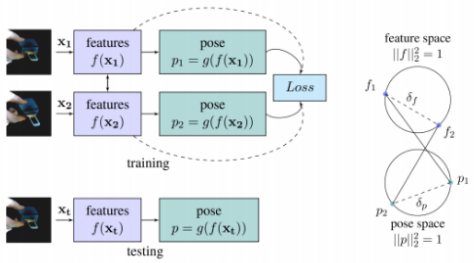

Figure 3: Method Described in the Domangolou Paper

2 Methods

For the first half of the project, we wanted to examine the methodology used in the Wohlhart paper [1]. This method appealed to us as an initial start, because it outlined much of the data wrangling and augmentation that we would need to do in order to be able to get many of these methods up and running.

We start by going through the application domain methodology before delving into how the theoretical results form the Wohlhart paper [1] addresses these application level concerns.

Current state of the art methods for estimating the 3D pose of objects involves using the Iterative Closest Point algorithm (ICP). This algorithm is described as follows from `https://en.wikipedia.org/wiki/3D_pose_estimation`:

1. Reconstruct projection rays from the image points
2. Estimate the nearest point of the projection ray to a point on the 3D contour
3. Estimate the pose of the contour with the use of this correspondence set
4. goto (2)

In this way the algorithm describes an iterative process to determining the pose. This can be time consuming if you do not have a strong initial estimate of the pose. It also can fail if there are occlusions in the image space.

Ideally we want an algorithm that will be robust to occlusions, fast to compute, and extensible in the application domain.

We are specifically considering the domain of the pick and place robotic challenge. In this domain, we have a limited set of objects that we want to be able to manipulate. This domain is quite a bit more constrained than other domains such as autonomous navigation in which there is a long tail distribution of objects. We have a very small tail that we can control. Because of these constraints, we are able to rely on having many views of the objects that we are considering.

For our initial method, we train on a single object with a wide variety of poses. Walhart [1] has shown that this methodology can be extended to recognizing object classes as well as poses.

In these kind of constrained application environments, a common practice is to create a database of each object in a series of different poses. Here, the intuition is that you can determine the pose up to some ϵ that is determined by the granularity of your database.

This methodology makes sense for the application domain that we are dealing in. For pick and place robotics, we conceivably have a set containing the objects that we are dealing with in current formulations of the problem domain.

The major problem for this method is that determining how similiar the data is from each other is non-trivial. With RGB or depth data alone, you are limited by many factors. Using something like the ICP algorithm as mentioned above could be costly and prone to complications with occluded data.

This method [1] looks at learning a descriptor to use as a key for finding the most similar pose in the database. The descriptor is generated using a convolutional neural network followed by two fully connected layers as shown in Figure 4.

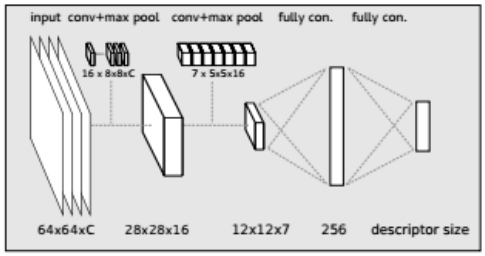

Figure 4: Network Architecture for the Wohlhart Method [1]

The activations of the last layer form the descriptor for the input image. This methodology works for RGB and RGB-D data.

The Walhart paper [1] outlines that a good mapping from the images to the descriptors should have that the Euclidean distance between the descriptors of the same object with the same pose be small and everywhere else be large. In this vein, a loss function is proposed that directly takes into account similarities. An interesting note from this algorithm is that the true pose does not need to be used during training time. It can strictly be used at validation and test time. The proposed loss function is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{triplets}} + \mathcal{L}_{\text{pairs}} + \lambda ||w'||_2^2$$

This consists of two loss functions and a regularizing term. The two loss functions can be explained further:

$$\mathcal{L}_{\text{triplets}} = \sum_{(s_i, s_j, s_k) \in \tau} c\left(s_i, s_j, s_k\right)$$

Where:

$$c\left(s_i, s_j, s_k\right) = \max\left(0, 1 - \frac{||f_w\left(x_i\right) - f_w\left(x_k\right)||_2}{||f_w\left(x_i\right) - f_w\left(x_j\right)||_2 + m}\right)$$

And $f_w\left(x\right)$ is the output to the neural network.

This term $\mathcal{L}_{\text{triplets}}$ is taken between two examples. s_i and s_j are similar, while s_k is different form the other two.

The pair term can be expanded as:

$$\mathcal{L}_{\text{pairs}} = \sum_{(s_i, s_j) \in \mathcal{P}} ||f_w(x_i) - f_w(x_j)||_2^2$$

$\mathcal{L}_{\text{triplets}}$ ensures that similar objects are given similar descriptors and dissimilar objects have dissimilar descriptors. The term $\mathcal{L}_{\text{pairs}}$ enforces that for two images of the same object and the same pose irregardless of the illumination and other noisy factors, we obtain similar descriptors.

In using this we have a descriptor that can be more efficiently generated (just a single single propagation method) than other methods. This descriptor is also shown to be robust to illumination and small occlusions. In this way, the method meets the criteria for application that we have outlined above.

During training time, we will be using clearer objects with less clutter, and during test time, we will test against slightly occluded objects. This is in line with the Wohlhard method [1].

With these descriptors, we will create a database of images and ground truth poses using the descriptors as the key. At test time, we will feed an image through the network, get a descriptor, use it as a key, and then get the closest ground truth pose from the closest match.

3 Preliminary Results

3.1 Neural Network - TensorFlow

The code that we have written can be found on our github page `https://github.com/colegulino/Bootstrapping-Object-Pose-Detection`

3.2 Data augmentation

Based on the description of the cost function, we require varying degree of similarity among poses of an object for the dataset. This usually is not available directly from the dataset. For sub-degree resolution in the pose of the object, we decided to develop a simulation framework, which could take the geometry (mesh) of the object as available in the dataset, and generate the RGB-D data at any arbitrary pose. This will give us the power to control the similarities in the poses as required by our method.

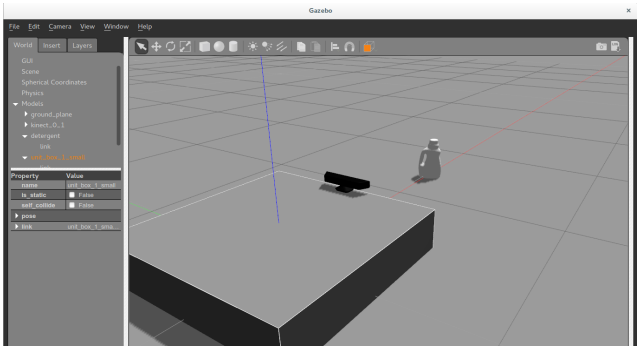

Figure 5: Gazebo simulation setup for generating RGB-D data

We developed an initial version of a Gazebo based simulation framework, which takes the mesh of the object as an input, and uses an RGB-D sensor framework to extract image and depth data and expose it using a ROS framework. The setup is shown in figure-5. An example RGB-D output pointcloud is shown in 6.

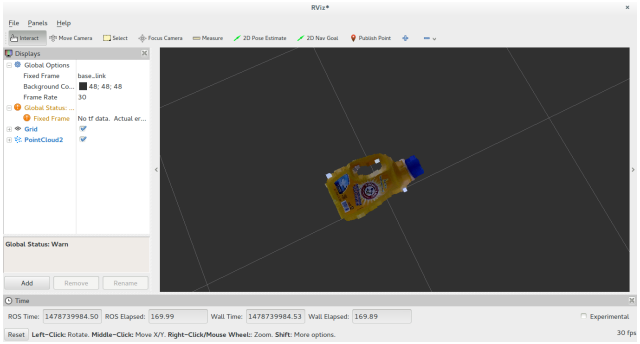

Figure 6: An example of the RGB-D pointcloud data for the detergent object

4 Final plan

The final plan for the finished project has not changed, but we have talked about a few other ideas in order to differentiate our method from the other methods that we have mentioned in the literature survey.

First and foremost we plan to further flesh out the methodology that we have stated above. We will also explore the other methods mentioned in the Introduction.

Finally, we are considering how we can use generative models in order to improve our application. One of the major issues with object detection in 3D point clouds is that we only see one face of the object. We want to explore the idea of using generative methods to learn the distributions for possible 3D shapes of objects in a particular domain for example a supermarket. Then, this might help in generating full 3D model of the object from only a partial 3d point cloud. We would be very interested to see if this improves object recognition using point clouds.

References

[1] Wohlhart, Paul, and Vincent Lepetit. "Learning descriptors for object recognition and 3d pose estimation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[2] Hinterstoisser, Stefan, et al. "Gradient response maps for real-time detection of textureless objects." IEEE Transactions on Pattern Analysis and Machine Intelligence 34.5 (2012): 876-888.

[3] Kehl, Wadim, et al. "Deep Learning of Local RGB-D Patches for 3D Object Detection and 6D Pose Estimation." European Conference on Computer Vision. Springer International Publishing, 2016.

[4] Doumanoglou, Andreas, et al. "Siamese Regression Networks with Efficient mid-level Feature Extraction for 3D Object Pose Estimation." arXiv preprint arXiv:1607.02257 (2016).

[5] Chopra, Sumit, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. IEEE, 2005.