Grasp Type Estimation for Myoelectric Prostheses using Point Cloud Feature Learning

Ghazal Ghazaei1,2, Federico Tombari2, Nassir Navab2 and Kianoush Nazarpour1,3

Abstract—Prosthetic hands can help people with limb difference to return to their life routines. Commercial prostheses, however, have several limitations in providing an acceptable dexterity. We approach these limitations by augmenting the prosthetic hands with an off-the-shelf depth sensor to enable the prosthetics to see the object’s depth. A single view (2.5-D) snapshot, and estimate an appropriate grasp type; using a deep network architecture based on 3D point clouds called PointNet. The human can act as the supervisor throughout the procedure by accepting or refusing the suggested grasp type. We achieved the grasp classification accuracy of up to 88%. Contrary to the case of the RGB data, the depth data provides all the necessary object shape information, which is required for grasp recognition. The PointNet not only enables using 3-D data in practice, but it also prevents excessive computations. Augmentation of the prosthetic hands with such a semi-autonomous system can lead to better differentiation of grasp types, less burden on user, and better performance.

I. INTRODUCTION

Losing a hand can cause inevitable limitations to an individual’s life. Prosthetic hands can provide such amputees with the opportunity of returning to their normal activities. However, control of these prosthetic hands is still unnatural and limited to a few degrees of freedom. Therefore, research is ongoing to further improve the functionality of prosthetic hands [1]–[6].

There are several research works which employ 2-D and 3-D visual data to boost the performance of prosthetic hands, demonstrating the benefit of using vision as an additional modality to the electromyogram (EMG) signals [5]–[10]. In [5], 10 consecutive 2D RGB snapshots of objects together with ultrasound distance information are used as an input to a rule-based reasoning algorithm to estimate among four different grasp types. Later in [11], fusion of different sensory data including myoelectric recordings, computer vision, inertial measurements and embedded prosthesis sensors (position and force) led to semi-autonomous and proportional control of a prosthetic hand in multiple DOFs. RGB-D imaging was used to estimate the shape, size and orientation of objects. Another work that benefited from 3-D sensors was [6], which proposed a combination of stereo-vision and augmented reality (AR) for better user interface and control of the hand. In [8], [9], an RGB image is fed to a two-layer convolutional neural network (CNN) [12] to choose the best grasp among four different types. The algorithm can effectively classify objects based on their appropriate grasp category without any additional sensor or measurement.

Most works indicate that the use of additional modalities such as depth can be beneficial in grasp estimation with vision. Despite the high grasp recognition performance in [8], [9], the system was sensitive to change in distance and view point, which can be overcome by using a depth sensor. Additionally, background removal is a challenging task in 2D images, while depth data can ease this procedure significantly and provide better outcomes. With the rapid development of 3-D computer vision and depth sensors, it is now much easier than before to use depth data and process it. There are a variety of sensors, which can provide depth for objects as close as 10 cm. They are adequately small (from 6-9 cm) and can be effectively integrated into the prostheses. Intel® RealSense™ D400 series1, REAL3™ image sensor family by Infineon Technologies AG2 and Pico depth sensors by Pmdtechnologies AG3 are among the 3-D sensors with the potential to be integrated into a prosthesis.

Although these innovations facilitate the use of depth sensors, 3-D data processing can be computationally expensive. A solution could be the PointNet [13] approach, that relies on a comparatively shallow network. PointNet uses point cloud data directly and has shown great performance on several tasks including 3-D shape classification, which is of our interest [13]. Another benefit is that the RGB data is no more required, which eliminates the use of unnecessary data and accelerates the performance. By employing the recent developments in computer vision, this paper tries to improve the grasping performance of artificial hands and presents an efficient semi-autonomous grasp estimation approach for a single view (2.5-D) point cloud (set of points), which can easily be implemented on an available artificial hand by a single depth sensor. That is, a depth sensor augmented on a prosthetic hand can capture a single view RGB-D images, while depth data can ease this procedure significantly and provide better outcomes. By employing the recent developments in computer vision, this paper tries to improve the grasping performance of artificial hands and presents an efficient semi-autonomous grasp estimation approach for a single view (2.5-D) point cloud (set of points), which can easily be implemented on an available artificial hand by a single depth sensor. That is, a depth sensor augmented on a prosthetic hand can capture a single view RGB-D images, while depth data can ease this procedure significantly and provide better outcomes.

This work is supported by UK Engineering and Physical Sciences Research Council (EP/R004242/1).

1 School of Engineering, Newcastle University, Newcastle-upon-Tyne NE1 7RU, UK.
2 Technical University of Munich, Munich, Germany.
3 Institute of Neuroscience, Newcastle University, Newcastle-upon-Tyne NE2 4HH, UK.

Emails: {g.ghazaei1}@newcastle.ac.uk

Workshop on Human-aiding Robotics
International Conference on Intelligent Robots and Systems 2018

image by a trigger command recorded from the amputee user. The image is processed, converted to point cloud and fed into the PointNet, which results in an automatic grasp act in the prosthesis by classifying the object based on its appropriate grip pattern. As geometric information is substantially important for grasp gesture of an object, we extract the normal vectors of point clouds and input that to the PointNet as extra information. Results indicate estimation improvement of $\sim 10\%$ when normal data is added.

II. METHODS

A. Dataset

1) Washington RGB-D dataset: There are numerous datasets, which exploit depth data. However, due to the specific aim of grasping in this paper, we focused on the ones that include graspable objects. Among those, a large RGB-D dataset collected at the University of Washington [14], [15] includes sufficient data and presents mostly graspable objects, for example those shown in Figure 1. There are RGB and depth images of 300 common everyday objects from multiple view angles (total: 250,000 RGB-D images) collected with a Microsoft Kinect\footnote{Microsoft Kinect \url{https://www.xbox.com/en-us/kinect}. Accessed: 2018-01-09}. We selected 3321 2.5-D point clouds, which are distributed in 48 categories and sampled from almost every 12 views per object in each object category. In some categories, some objects may not be used to avoid unnecessary repetition.

We used the processed point clouds provided in [15] as the background is already removed. We manually labeled these objects based on their appropriate grasp type following the process adopted in [8], [9] to four different grasp groups: tripod, pinch, palmar wrist neutral and palmar wrist pronated.

2) BigBIRD ((Big) Berkeley Instance Recognition Dataset): Since the Washington RGB-D dataset includes more objects in palmar wrist neutral and palmar wrist pronated than tripod and pinch grasp categories, we added some data from the BigBIRD dataset [16] to compensate and provide sufficient diversity in every grasp class. It includes 100 objects and 600 RGBD images for each. Here, we picked 12 categories, including 656 2.5-D point clouds sampled from processed files (background is removed). The RGB and depth data are collected by Canon and Primense Carmine cameras. Figure 2 depicts the objects selected from the BigBIRD.

Combining these two datasets, Table I indicates the number of objects per grasp class in each group. It is worth noting that sample selection per available views rate in each object category in the BigBIRD is 5 times more than the Washington RGB-D dataset due to the presence of many similar objects in the latter. In this way, the overall number of point clouds per grasp is nearly uniformly distributed as shown in the last row of Table I.

Table I: A summary of data distribution within grasp groups and datasets. Palmar WN and Palmar WP stand for palmar wrist neutral and palmar wrist pronated grasps respectively.

<table>
<thead>
<tr>
<th>Grasp Pattern</th>
<th>Dataset Type</th>
<th>Pinch</th>
<th>Palmar WN</th>
<th>Tripod</th>
<th>Palmar WP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Washington</td>
<td>objects</td>
<td>62</td>
<td>80</td>
<td>15</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>point clouds</td>
<td>740</td>
<td>956</td>
<td>644</td>
<td>981</td>
</tr>
<tr>
<td>BIGBIRD</td>
<td>objects</td>
<td>4</td>
<td>-</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>point clouds</td>
<td>183</td>
<td>-</td>
<td>473</td>
<td>-</td>
</tr>
<tr>
<td>Combined</td>
<td>objects</td>
<td>66</td>
<td>80</td>
<td>23</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>point clouds</td>
<td>923</td>
<td>956</td>
<td>1117</td>
<td>981</td>
</tr>
</tbody>
</table>

B. Data preparation

Although the point clouds were already processed and the background was removed properly, more processing was required as the PointNet requires the point clouds to be zero-mean and normalized into an unit sphere. As an additional data relevant to object geometry, the surface normals for each point cloud were also estimated. Although there are several normal estimation methods available, one of the simplest approaches is to approximate the normal to a point on the surface by estimation of the normal of a plane tangent to the surface, which becomes a least-square plane fitting estimation problem. Consequently, the surface normal estimation problem is reduced to an analysis of the eigenvectors and eigenvalues of a covariance matrix created from the nearest neighbors of the query point. That is, for each point $\mathbf{p}_i$, the covariance matrix $\mathbf{C}$ can be calculated.
according to equation 1.

\[ C = \frac{1}{k} \sum_{i=1}^{k} (\mathbf{p}_i - \overline{\mathbf{p}}) \cdot (\mathbf{p}_i - \overline{\mathbf{p}})^T, \quad C \mathbf{v}_j = \lambda_j \mathbf{v}_j, \quad j \in \{0, 1, 2\} \]

where \( k \) indicates the number of point neighbors considered in the neighborhood of \( \mathbf{p}_i \) (here \( k = 100 \) provided us with desirable results), \( \overline{\mathbf{p}} \) illustrates the 3-D centroid of the nearest neighbors, \( \lambda_j \) represents the \( j \)-th eigenvalue of the covariance matrix, and \( \mathbf{v}_j \) the \( j \)-th eigenvector [17].

Finally, we uniformly sampled 2048 points for each point cloud. It is worth noting that we did not use the RGB data as it does not include any shape relevant information and consequently barely any grasping relevant data. Thus, each point cloud is composed of six coordinates \((x, y, z, n_x, n_y, n_z)\), where \( n_i \) represents the normal vector for \( x, y, z \). For training, point clouds are augmented by random rotation along the up-axis and jittering the position of each point by a zero-mean Gaussian noise (standard deviation, 0.01).

C. PointNet

Deep learning methods have shown great success in various classification tasks [18], [19]. Although point clouds are simple and unified type of geometric data structure and easy to learn from, they are not directly fed to a deep network architecture due to their irregular format. PointNet however can simply use point clouds as the input representation [13] thanks to its unique design (Figure 3).

Since a point cloud is a set of unordered 3-D points, PointNet requires certain symmetrization in the feed-forward computation and further invariances to rigid motions may also be needed. The main feature of PointNet is the presence of a single symmetric function called max pooling that aggregates the information from each point leading to invariance to input permutations. As shown in Figure 3, the network selects informative points of the point cloud during training in the first MLP (multi-layer perceptron) layers. These learned optimal values are accumulated into the global descriptor by the final fully connected layers. For our specific task of grasp estimation, this global descriptor should include particular distinctive features that represent each grasp category.

Batch normalization [20] and ReLU (rectified linear unit) are used for all the layers. We used the learning rate of 0.001 to train the network on an Nvidia GeForce GTX 960M GPU.

III. RESULTS AND DISCUSSION

We had a total of 3797 point clouds of which we used 80% for training, 10% for validation and the remaining 10% for testing. We trained the PointNet in two ways: 1) basic model including \((x, y, z)\) data only and 2) Extended model including surface normals, \((x, y, z, n_x, n_y, n_z)\). Results are depicted in Table II. Figure 4 illustrates the training curves for both models in the second fold of cross-validation. The extended model converges in fewer steps to a higher accuracy while taking longer training time.

![Figure 4](image_url)

According to table II, the results indicate about 79% average accuracy for the basic model and 85% average accuracy for the extended model. The procedure of processing an image and predicting relevant grasp for it takes about 0.03 seconds. It can be seen that using surface normals as additional coordinates is beneficial to the grasp estimation task (performance improvement up to \(\sim 10\%\) in one of the cross-validation folds). It seems to be a plausible claim as surface normals can provide more data relevant to the object shape and grasping gesture.

It can be observed that since the palmar wrist neutral grasp type includes the most distinctive types of objects compared to other grasp groups (objects that their length along \(y\)-axis is larger than their length along \(x\)-axis), the objects suitable for this grasp type are recognized with the highest accuracy. Moreover, the pinch grasp consisting of the least amount of data represents the lowest recognition accuracy. These results also fit with previous results reported in [8], [9].

Figure 5 indicates the confusion matrices of the second validation fold for both basic and extended models. As results
already presented, the extended model indicates a better distribution around the diagonal.

Some samples of incorrect grasp classification are demonstrated in Figure 6. It can be noticed that some errors are happening due to segmentation problems or depth data noise.

![Image of object point clouds from different views](image)

Fig. 6: Samples of object point clouds from different views led to incorrect grasp classification. hint: captions are ordered as T-P, where T and P represent true and predicted labels respectively.

**IV. Conclusion**

In this paper, an effective and efficient approach for augmenting a hand prosthesis with a depth sensor was presented. Compared to RGB data, depth data provides more shape and grasp relevant information and a depth sensor can be easily mounted on an artificial hand. We added further shape information through estimating surface normals, which led to better grasp estimation performance. Additionally, object segmentation is easier when using depth data.

There are several barriers in working with the depth data, namely noisy sensor output and extensive computations. The latter can be eliminated by the use of PointNet, which avoids excessive computations by using point clouds and an efficient architecture. The problem of noise can be overcome by utilizing recent 3-D sensors and noise removal algorithms. Still, depth sensor technologies are improving increasingly and they can be used for commercial artificial hands in near future.

**References**


