Transferring PointNet++ Segmentation from Virtual to Real Plants

Ayan Chaudhury[1,2], Peter Hanappe[3], Romain Azaïs[1,2], Christophe Godin[1,2], and David Colliaux[3]
[1]INRIA Grenoble Rhône-Alpes, team MOSAIC
[2]RDP Lab, ENS de Lyon, Univ Lyon, UCB Lyon 1, CNRS, INRA
[3]Sony CSL Paris
{ayan.chaudhury,romain.azais,christophe.godin}@inria.fr, {hanappe,colliaux}@csl.sony.fr

Abstract

One of the biggest challenges of deep neural network to perform segmentation of point clouds is the requirement of large amount of annotated data, which is expensive in terms of manual labour and time. For the case of plants, there is a scarcity of datasets to train the networks to perform organ segmentation for automated phenotyping applications. In this work, we explore how the use of virtual plants as modelled by stochastic L-systems can circumvent this problem. We investigate the effect of point density and how the complexity of the plant model affect the transfer of learning from virtual to real plants on a segmentation task based on PointNet++.

1. Introduction

Organ segmentation in 3D plant point cloud data is a fundamental problem in agricultural automation and plant phenotyping research. With the recent breakthrough of deep learning technology for point clouds [6, 7], a new research avenue has opened on applications of point cloud segmentation and recognition. Several strategies have been proposed to apply neural networks to point clouds [3]. However, a major challenge of applying general deep learning techniques in any particular domain is the availibility of large number of annotated data. Although there is an abundance of labelled point cloud data on regular objects [12] and urban scenes [4], there is a scarcity of labelled datasets for the case of plants. A common approach to handle the data scarcity problem is to generate synthetic data which mimicks the original scene/object. This approach has been shown to be promising in 2D [8], sometimes along with fine tuning the network using a small set of annotated real data [9]. However this type of approach has rarely been reported for plants until recently [10]. For the case of plants, the problem is more challenging due to the complex and self recursive geometrical structure with a large number of variability within the species. Generation of virtual plant models using L-system rules [5] has been a practice in the computer graphics and geometric modelling community for decades. Recently, it has been shown that labelled point cloud data can also be generated from L-system models [1]. In this work, we study whether L-system models have the potential to train deep networks to perform well in organ segmentation on real data. We also investigate what are the features that are critical for knowledge transfer from synthetic data to real data without retraning on real data (a.k.a. *fine tuning* in transfer learning literature).

Recently, only a couple of previous approaches investigated deep learning based segmentation of 3D point clouds of plants ([2], [10]). Ghahremani *et al.* [2] performed Wheat ear segmentation of point cloud data based on the strategy of PointNet [6] architecture, where manually annotated real data were used for training. Turgut *et al.* [10] performed a comparsion of state-of-the-art deep networks for segmenting Rosebush plants. In this work, each plant is represented as a large point cloud and the segmentation is performed on small patches of the point cloud containing few thousand points. The network thus segments small portions of leaves, branches or flowers. In this case, the local geometric structure of a point gives a good description of the semantic class of a point. In contrast to this work, we question whether the whole plant can be segmented in one shot, without subdiving the point cloud into small patches.

2. Generation of the dataset

2.1. Synthetic data

We followed the L-system modelling strategy as proposed in [1] using their publicly available code. In this work, we considered Arabidopsis plant for our experiments. We developed an L-system model of Arabidopsis where we assigned unique label to each organ of the plant (we used 4 labels: fruit, pedicel, stem, and leaf) in the rules of the procedural model. The scale of the plant and the parameters of the model are designed on the basis of observations from real plants. Typical examples of the parameters include length of fruit, bending of stem, number of organs, radius of each branch, etc. We also introduced stochasticity in the parameters, so that every execution of the model can produce a different variety of the plant under consideration. This is achieved by sampling the parameter value from a Gaussian distribution with a mean of average value in real plants and a standard deviation of possible variabilities. We focus on specific geometric traits of plant growth which we believe are crucial for mimicking the real data, e.g. shape, size, length, curvature and orientation of the organs. For example, bending of the stem is an important parameter, which we model as a spline curve with controllable stochastic elasticity factor. This allows us to model a large variety of bendings that can render realistic data. Since the real plants have large variabilities with the branch thickness, the training data should include large varieties of branch radius. We model this effect in the virtual plant by sampling the value of radius from a uniform distribution of probable true values. For leaves for instance, we found it essential to model precisely the few tiny leaves on the main stem that have shape, size and orientation different from that of the leaves in the rosette.

To test the robustness of our approach, we analyzed the effect of different modifications of our protocol. First, we generated models of plants with and without small leaves in order to demonstrate the added complexity of training when leaves are present (discussed in results section). Second, we changed the density of points, which is seen to play a crucial role in training.

2.2. Real data

Real data was acquired using a robotic platform with an arm moving a camera around the plant. From a set of 72 views, the 3D point cloud of the plant was reconstructed using a space carving technique. We follow the acquisition and reconstuction method similar to [11] using their publicly available code. The dataset of 13 point clouds was then annotated manually using the CloudCompare software.

2.3. Training

The model for training synthetic data is initially rendered as a mesh using .obj file format. We sampled 4096 points on the surface of the mesh. Points are shuffled as a preprocessing step so that the network learning is invariant to the order of the points. We generated 1000 virtual plants which were split into 900/80/20 for training, validation and test. We used the PointNet++ architecture as introduced in [7] without batch normalization (that did not bring any improvement).

3. Results

We used here the standard mean intersection over union metric to evaluate the result as, mIoU = TP/(TP+FN+FP).

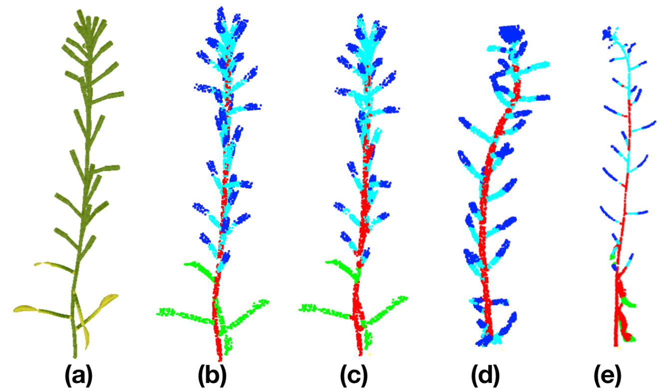

Figure 1. Virtual plant, ground truths and segmentation results of Arabidopsis thaliana. Sample of a virtual plant model (a), corresponding ground truth point cloud (b), corresponding segmentation result (c), segmentation result on real data without leaf (d), and with leaf (e).

3.1. Results on the test virtual plants dataset

We first considered the segmentation result of the network trained on virtual plants when applied to test on virtual plants (Fig. 1c). We obtained a mIoU of 0.77 for the stem, 0.90 for fruits, and 0.85 for pedicels. This shows that in principle the PointNet++ provides promising results when applied to complete point clouds.

3.2. Preliminary results on real plants

We then considered a first application of our method to real plants (Fig. 1d, e), with the aim of analyzing how to improve our virtual plants for training so that they optimize the real plant recognition rates. In a preliminary experiment, we could obtain a maximal mIoU of 0.56, suggesting that further improvement of our virtual plants is required. We therefore are currently refining our construction of virtual plants so that mIoU is improved on real plants. In particular, we address the bending of axes that needs to be more realistic with respect to the observed real plants and the number of organs per plant.

3.3. Effect of the sampling strategy

We tested 2 sampling strategies to generate point clouds from the mesh of virtual plant model for training. In the first strategy, we generated a point cloud via Poisson disk sampling of the triangle mesh [13]. In the second method, we used the method proposed in [1] to obtain uniform density. For both cases we performed testing using PointNet++. The mIoU averaged over 20 plants is 0.68 with the first strategy whereas it is 0.84 in the second strategy, showing the importance of using uniform point clouds in our pipeline.

Acknowledgement: This work is supported by Robotics for Microfarms (ROMI) European project.

References

[1] A. Chaudhury, F. Boudon, and C. Godin. 3D plant phenotyping: All you need is labelled point cloud data. In *CVPPP - ECCV*, pages 244–260. Springer, 2020. 1, 2

[2] M. Ghahremani, K. Williams, F. M. K. Corke, B. Tiddeman, Y. Liu, and J. H. Doonan. Deep segmentation of point clouds of wheat. *Frontiers in Plant Science*, 12:429, 2021. 1

[3] Y. Guo, H. Wang, Q. Hu, H. Liu L. Liu, and M. Bennamoun. Deep learning for 3D point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1

[4] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-1-W1, pages 91–98, 2017. 1

[5] A. Lindenmayer and P. Prusinkiewicz. *The algorithmic beauty of plants*, volume 1. New York: Springer-Verlag, 1990. 1

[6] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[7] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5105–5114, 2017. 1, 2

[8] Y. Toda, F. Okura, J. Ito, S. Okada, T. Kinoshita, H. Tsuji, and D. Saisho. Training instance segmentation neural network with synthetic datasets for crop seed phenotyping. *Communications biology*, 3(1):1–12, 2020. 1

[9] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977, 2018. 1

[10] K. Turgut, H. Dutagaci, G. Galopin, and D. Rousseau. Segmentation of structural parts of rosebush plants with 3D point-based deep learning methods. *arXiv preprint arXiv:2012.11489*, 2020. 1

[11] T. Wintz, D. Colliaux, and P. Hanappe. Automated extraction of phyllotactic traits from arabidopsis thaliana. In *CVPPP*, 2018. 2

[12] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 1

[13] C. Yuksel. Sample elimination for generating poisson disk sample sets. In *Computer Graphics Forum*, volume 34, pages 25–32. Wiley Online Library, 2015. 2