



HAL
open science

Toward a Procedural Fruit Tree Rendering Framework for Image Analysis

Thomas Duboudin, Maxime Petit, Liming Chen

► **To cite this version:**

Thomas Duboudin, Maxime Petit, Liming Chen. Toward a Procedural Fruit Tree Rendering Framework for Image Analysis. 7th International Workshop on Image Analysis Methods in the Plant Sciences, Jul 2019, Lyon, France. pp.4 - 5. hal-02140367

HAL Id: hal-02140367

<https://hal.science/hal-02140367>

Submitted on 7 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Toward a procedural fruit tree rendering framework for image analysis

Thomas Duboudin¹, Maxime Petit¹, and Liming Chen¹

¹LIRIS, CNRS UMR 5205, Ecole Centrale de Lyon, France

Abstract

We propose a procedural fruit tree rendering framework, based on Blender and Python scripts allowing to generate quickly labeled dataset (*i.e.* including ground truth semantic segmentation). It is designed to train image analysis deep learning methods (*e.g.* in a robotic fruit harvesting context), where real labeled training datasets are usually scarce and existing synthetic ones are too specialized. Moreover, the framework includes the possibility to introduce parametrized variations in the model (*e.g.* lightning conditions, background), producing a dataset with embedded *Domain Randomization* aspect.

Keywords: Synthetic Fruit Dataset, Harvesting Robotics, Procedural Model, Domain Randomization

1 Introduction and Previous Work

State-of-the-art methods for object recognition and grasping are currently mostly based on deep neural network. Despite tremendous results, these methods require a huge amount of labeled data in order to be trained. This is a major drawback in the field of robotics fruit-harvesting, where existing dataset are usually too small and/or dedicated to a single specie (*e.g.* [1, 2]).

We previously tackled this issue for indoor object grasping robots using a simulated environment and rendering engine [3, 4]. We apply here a similar strategy for fruits harvesting problems by defining a framework capable of generating scenes of fruit trees coupled with procedural scripts controlling parameters (*e.g.* position of the fruits, type of background, lightning condition) to introduce realistic variations for outdoor data¹.

One of the most photo-realistic synthetic dataset of fruit trees is the work of Barth *et al.* [2] for sweet pepper. This come at the price of a huge computational cost (10 min/frame with a 16 core processor) preventing it to be easily extendable in order to produce large dataset for other fruits. In fact, such high degree of photo-realism does not seem to be needed for deep simulated learning [5]. That is why we aim at only an adequate photo-realism with quick rendering and easy-generation method, allowing scientists to create their own fruits or tree dataset according to their precise research interests.

2 Material and Methods

We chose the open-source Blender as the 3D-modelling and rendering software. Every options accessible through the Blender GUI can be reached and modified through a Python API, enabling us to entirely control the simulation with scripts.

The framework can be decomposed in two steps : first the generation of the tree or plant models, then a rendering script.

¹Source code available at <https://github.com/tuboudi/IAMPS2019-Procedural-Fruit-Tree-Rendering-Framework>



Figure 1: Samples of synthetic oranges (top) and apples (bottom) with different lightning conditions and rendering qualities.

It mainly contains a animation loop, such as that each time-step is responsible for the rendering of one image (and a semantic segmentation map, corresponding to the ground truth label), different from the previous ones.

Tree models are generated following the rules of Weber and Penn [6] using an existing Blender tree generation add-on, controlled by the Python generation script. It defines a number of parameters such as the branching frequency, the decrease in the radius of trunks and branches, the overall direction of the branches (up or down), the ratio tree height/branches length, *etc.* The fruits and the leafs are randomly and uniformly added upon the naked tree following defined densities.

The camera movements has to be defined in the rendering script, in which position and orientation of the camera should be directly provided at each time-step. We usually use a new random position and orientation at each iteration, such that the camera is globally pointing toward the models (see Fig.2). While it has to be noted that the camera trajectories do not have

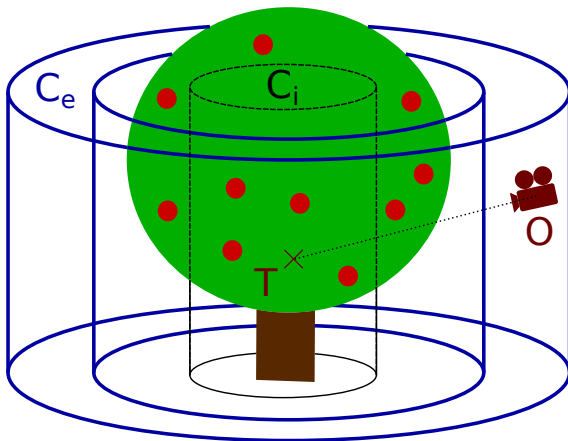


Figure 2: Schema of the camera point of view generation. Camera origin O and target point T are uniformly picked from respectively the hollow external cylinder C_e and the internal cylinder C_i (centered around the model).

to be continuous, we can also create robotic-like continuous trajectories toward branches on a tree. In addition, depth of field and rendering-engine parameters (e.g. number of rays for the ray-tracing engine) can be randomly modified within a certain range every time-step.

Background and lightning are also controlled in the rendering script, and can be easily changed every few time-steps to generate diverse images. This is a much needed feature allowing *Domain Randomization*, known to be efficient in reducing the *Reality Gap* encountered when transferring deep neural networks from simulation to the real world [7]. In order to have complex backgrounds we use freely available spherical HDRIs (high-dynamic-range 360° images), which control the lightning and the background objects. The tree models are placed in the center of the HDRIs.

3 Results and Discussion

Overall, the rendering of one pair of images (512x512 pixels, raw and ground truth for semantic segmentation) is fairly quick with $\sim[10s:30s]$ with GPU-rendering (NVIDIA GTX 1080). Fig.1 illustrates different possible variations for the scene generation (e.g. fruits, light) and the quality of the rendering.

4 Conclusion and Future Work

We designed a framework for semantic segmentation and object recognition field related to robotics fruit harvesting problems. It allows a quick and efficient fruit trees scene generation with parametrized variations, thus producing labelled images embedding a *Domain Randomization* aspect.

We aim to extend our framework to reproduce the variations occurring during the lifetime of the fruits and tree. For instance,

an implementation of the fruit senescence and decay [8] unlocks the training of mature fruit picking robot and disease detection systems. We also plan to improve the photo-realism of the simulated image by using a Generative Adversarial Network as in [9], allowing to keep the rendering time needed quite low.

Acknowledgment

This work is supported by the french National Research Agency (ANR), through the ARES labcom (grant ANR 16-LCV2-0012-01) and by the CHIST-ERA EU project "Learn-Real".

References

- [1] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "Deepfruits: A fruit detection system using deep neural networks," *Sensors*, vol. 16, no. 8, p. 1222, 2016.
- [2] R. Barth, J. IJsselmuiden, J. Hemming, and E. J. Van Henten, "Data synthesis methods for semantic segmentation in agriculture: A capsicum annum dataset," *Computers and electronics in agriculture*, vol. 144, pp. 284–296, 2018.
- [3] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2018, pp. 3511–3516.
- [4] M. Petit, A. Depierre, X. Wang, E. Dellandréa, and L. Chen, "Developmental bayesian optimization of black-box with visual similarity-based transfer learning," in *The 9th Joint IEEE Int. Conf. on Development and Learning and on Epigenetic Robotics (ICDL-Epirob)*, 2018.
- [5] M. Rahnemoonfar and C. Sheppard, "Deep count: fruit counting based on deep simulated learning," *Sensors*, vol. 17, no. 4, p. 905, 2017.
- [6] J. Weber and J. Penn, "Creation and rendering of realistic trees," in *Int. ACM Conf. on Computer graphics and interactive techniques*, 1995, pp. 119–128.
- [7] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.
- [8] J. T. Kider Jr, S. Raja, and N. I. Badler, "Fruit senescence and decay simulation," in *Computer Graphics Forum*, vol. 30, no. 2, 2011, pp. 257–266.
- [9] R. Barth, J. Hemming, and E. J. van Henten, "Improved part segmentation performance by optimising realism of synthetic images using cycle generative adversarial networks," *arXiv preprint arXiv:1803.06301*, 2018.