From Perspective View to Bird's Eye View in Agricultural Environments

Simon Rumley, Lucas Teixeira, Margarita Chli
Vision For Robotics Lab, ETH Zürich, Switzerland
Anastasia Thoma, Paul Beardsley, Unity Technologies

***Abstract*— With the emergence of robots driving or flying under the canopy of agricultural environments, localization becomes a problem given that both GPS and traditional, sparse feature-based place recognition perform poorly in such environments. This paper proposes an approach, which converts imagery from an agricultural orchard taken at the below-canopy level into bird's-eye-view imagery, essentially generating a top view of the field indicating the tree positions in the horizontal plane. This is a step towards registering low- and high-altitude imagery. Existing state-of-the-art learning-based methods for such tasks, known as Perspective View to Bird's Eye View (PV2BEV) exist for urban scenes, particularly in the self-driving vehicle domain. Here, existing methods are evaluated in an agricultural setting, which poses notable challenges due to lack of structure and variability. We create high-quality synthetic datasets for the training of networks. We show preliminary evaluations on both synthetic and real imagery.**

I. Introduction

Agricultural robots are being developed for the full pipeline of operations, from monitoring through sowing, weeding, spraying, and harvesting [1]–[4]. Agriculture sensing in the past has been dominated by remote sensing, starting with satellite imagery and increasingly now over-canopy drone imagery. However the appearance of autonomous ground robots or low-level drones that work under the canopy, as well as the ubiquity of phone cameras, raises the question of how under- and over-canopy imagery can be matched, see Figure 1.

This registration is useful in a scene model that integrates data from under- and over-canopy sensors, such as in a Digital Twin [5]. For example, autonomous agricultural robots can register live imagery to an apriori map, to retrieve or update mission planning data. Or a farm operator can use phone imagery to add details to a drone dataset, say to provide high-resolution ground images showing fruit development to supplement the aerial view.

One way to attack this problem is to localize the ground/low-altitude robot accurately enough that ground imagery can be registered directly with an aerial view in the same coordinate frame. But this is not straightforward - GPS has insufficient accuracy at a few meters, while DGPS is accurate to a few centimeters but is expensive, and for computer vision a farm environment like an orchard has few distinctive fixed features plus there is temporal change due to growth. This motivates our work on using ground-level imagery to synthesize a bird's eye view (this paper is focused specifically on generating a bird's eye view, but we mention the matching task in order to provide context).

This work was supported by Unity Technologies.

Bird's eye view images are useful also in the case where there is no aerial imagery - path planning for a robot is simpler in a 2D top-view, and the notions of distance and position are more easily computable and understandable in this representation; and for tele-robotics including remote operation or remote oversight, the top-view is more easily comprehensible to a remote human operator.

Methods for transforming a perspective view to a bird's eye view, termed PV2BEV, are recent developments, especially in the domain of self-driving cars [6]. Our **contribution** is two-fold - firstly to investigate PV2BEV for an agricultural environment as opposed to an urban environment, and secondly to describe a case study of generating a synthetic dataset of trees, training a PV2BEV model using Deep Learning, and applying it to both synthetic and real-world datasets.

(a) Under-canopy view (b) Over-canopy / Bird's eye view

Fig. 1: Our test site in the winter

II. Related Works

PV2BEV research is mainly focused on autonomous cars. The paper proposed by Ma et al. [6] gathers most of the recognized studies on this topic. Many methods often require the use of several pieces of equipment such as 360 cameras, Lidar, IMU, and others. The goal of this project was to find a technique that required the use of only one camera and no other equipment. The purpose was also to be able to install, adapt and test several methods using different strategies to obtain a BEV from a PV of an agricultural scene. With these constraints, the research quickly converged on 3 different methods called Monolayout [7], Projecting Your View Attentively [8] and MonoDETR [9].

The first method, called Monolayout [7], is homography based. Its network is built to estimate the amodal scene layout. The amodal scene layout refers to the 3D arrangement of objects and their spatial relationships in the scene, including both visible and occluded objects that are not fully visible in the image. In the paper by Mani et al. [7], they proposed an architecture where there are two main stages. In

the first stage, the network generates a rough amodal layout estimation using an encoder-decoder architecture. It consists of a context encoder that extracts multiscale features, an amodal scene decoder which decodes the shared context to produce an amodal layout of the scene, and a dynamic scene decoder that predicts the vehicle occupancy. In the second stage, it uses two discriminators to ensure that the predicted static layouts align with the expected distribution of road geometries and ground-truth vehicle occupancy. The main advantage of this method is that it allows for estimating the shape of partially occluded objects. The input of the network is a perspective image taken by a camera facing the front of the vehicle. The output of the network is a segmented bird's eye view image that represents the layout of vehicles in traffic in front of the autonomous car.

The second method is called Project Your View Attentively [8] and is MLP based. This method takes the same input images as Monolayout and also produces the same type of output (top-view segmented images). The PYVA method is based on a GAN framework. It first extracts features from the input image using an encoder (ResNet). Then a cross-view transformation module enhances the features for view projection. In the end, a decoder is used to produce the top-view mask. The cross-view transformation module allows for predicting the camera's elevation and heading angles. It is used to project the image features onto a top-down bird's-eye view representation of the road. This module improves reprojection by an attention-based refinement. It refines the initial bird's eye view representation of the scene by focusing on the most relevant regions. The advantage of this method is that it is more precise in defining the visible (non-occluded) parts of the objects.

The third method is called MonoDETR [9] and uses a transformer architecture. It adopts a depth-guided scheme for aggregating features from the depth-related regions in the global context. This scheme estimates the density distribution of objects in a scene from a single image. The main advantage of this method is that it can detect objects of different sizes and shapes, as well as partially visible or hidden objects.

However, this method has some disadvantages in terms of the output format. The output of MonoDETR represents the objects (detected cars) by the coordinates of their 3D bounding boxes. To train the method, it is necessary to know the bounding boxes of the objects in the dataset. Their coordinates can be complicated to extract, even when generating a dataset by simulation. Moreover, this output representation is different from the two methods described above. To compare the MonoDETR method to PYVA and Monolayout, it would be necessary to create an aerial view image based on the coordinates of the bounding boxes, which is also a long task to implement. Another disadvantage of this method is that it takes about eight times longer to train than Homography and MLP-based methods.

III. Simulation-based Data Generation

The generation of simulated data for agricultural environments is the foundation of our work. In order to simulate agricultural ecosystems, such as orchards and crop plants, we created a rendering framework. It makes use of the following elements:

- Cinema-quality synthetic tree and plant models from SpeedTree [10]
- Topographic information from SwissTopo [11]
- Various small objects to add naturality, from online 3D model sources
- An in-house Vulkan-based render that can handle large-scale scenes

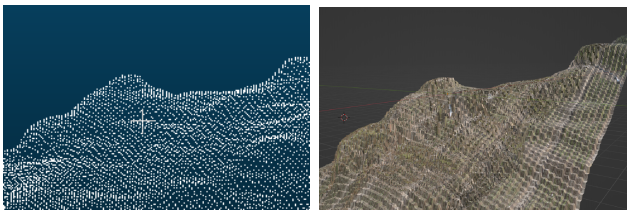
The generated dataset consists of a rendering of the 3D model for a ground-level view and a binary image for the bird's eye view. For an example tree model, see Figure 2.

- Several different species of tree were used including apple, cherry, and almond.
- Several times of year were used for each tree to model its appearance (varying foliage) in different seasons.

Fig. 2: SpeedTree apple trees: close-up, summer and winter.

The tree models are input to an automatic process that distributes trees on a topographic terrain captured from the real-world. Adjustable scene parameters are the type of plants, the size of the crop, the spacing between plants, the type of terrain, the presence of accessories faithful to real-world farm settings such as stones on the ground, and randomness in the placement. The arrangement of the plants follows a rectangular alignment similar to Figure 1. Figure 3 shows an example terrain.

The system generates large image datasets of the perspective view and the binary bird's eye view of the scene automatically. These images provide the input for training selected PV2BEV methods.



(a) Point cloud of a mountain (b) 3D object based on real topology

Fig. 3: Real topology extraction for simulation

IV. Experiments

We investigated the Monolayout, PYVA, MonoDETR methods for agricultural scenes. The MonoDETR method was eliminated due to disadvantages of the representation and long training times, as discussed in Section II. Monolayout and PYVA were evaluated according to the expected challenges that are present in agriculture, namely variations due to season change, variations due to different species of trees, and variations due to spacing between plants. We report mean Average Precision (mAP) and mean Intersection over Union (mIoU) for our experiments.

A. Comparative analysis for Monolayout and PYVA

Firstly, the effect of different sizes of training sets was investigated, as shown in Table I.

Because PYVA produced the best results with small datasets and has a good precision/training time ratio, it was chosen to continue with further experiments.

dataset size	Monolayout	PYVA
400 images	0.1095	0.6318
1000 images	0.2964	0.6912
3200 images	0.6536	0.6488

TABLE I: mIoU scores of trained networks with different dataset sizes, for the two algorithms.

B. Influence of the season

For this experiment, the PYVA network was trained with images of an apple orchard in the summer. Testing was then done on imagery with a different tree configuration and for all of the seasons. Figure 4 shows qualitative results and Table II (Summer set) shows quantitative ones obtained for the case when the network was trained with summer apple trees, and tested on summer apple trees with different tree geometry, scene accessories, skies and terrain. Figure 5 shows results for the case when the training set contains summer apple trees and the test set contains autumn apple trees. *For Figures 4,5,6, and 7, the color image is the input image. The ground-truth is at the top-right and the prediction is at the bottom-right. The white areas represent the trees and the black ones the other parts of the scene, mainly the ground. For the real input images, only the prediction is available at the right.*

Fig. 4: Prediction/Test of the PYVA method on an apple orchard in **summer**. The network was trained with 1200 images of an apple orchard in summer.

Test Set Season	mIoU	mAP
Summer	0.50	0.55
Spring	0.47	0.53
Autumn	0.48	0.59
Winter	0.36	0.42

TABLE II: mIoU scores of a network trained on apple trees in the Summer and tested on apple trees during different seasons

Fig. 5: Prediction/Test of the PYVA method on an apple orchard in **autumn**. The network was trained with 1200 images of an apple orchard in summer.

Based on the qualitative and quantitative results in Table II, the difference in textures, between a training set and a testing set does not seem to influence the quality of the prediction. We can therefore conclude that the difference in season, and therefore in textures, does not influence the quality of the prediction. However, the loss in foliage in winter does degrade the performance probably due to the change in geometry.

C. Influence of the type of tree

For this experiment, the PYVA network was trained using an apple tree dataset but the test set consisted of cherry trees. Figure 6 shows an example result. The cherry trees have sufficiently different geometry and appearance from apple trees that the prediction fails.

Fig. 6: Prediction/Test of the PYVA method on a **cherry** orchard in summer. The network was trained with 1200 images of an apple orchard in summer.

D. Influence of the spacing between trees

The spacing between trees is closely related to the amount of occlusion which will occur. For this experiment, we investigated the effect of changing space. The trees in Figure 4 are spaced at 14[m]x16[m] apart, and the trees in Figure 7 are spaced at 12[m]x14[m]. The mIoU and the mAP on a scene with wider tree spacing are 0.50 and 0.55, respectively,

while 0.46 and 0.50 on the one with narrower tree spacing. This indicates that the PYVA method is adversely affected by inter-tree occlusion, and the topic would benefit from further investigation.

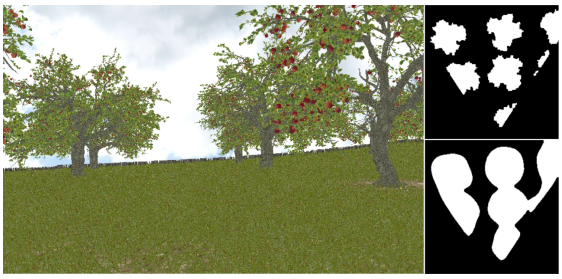


Fig. 7: Prediction/Test of the PYVA method on a summer apple orchard with **12[m]x14[m]** plant spacing. The network was trained with 1200 images of an apple orchard in summer, with the same spacing.

V. Experiments on real images

This last experiment is based on real images. The goal is to see how well the PYVA method, trained on synthetic images, was able to predict a bird's eye view of real images without domain adaptation.

The PYVA was trained with a dataset consisting of synthetic images of orchards over the four seasons, with a validation set that contained only synthetic images of apple trees in winter. This validation set was chosen to correspond to the case of the real images, which are of apple trees in winter 2022-23. Figures 8, 9, and 10 show qualitative results (ground-truth is not available). The algorithm is successful at predicting tree configuration and spacing. Tree shape is poor in the prediction since the real-world trees are all approximately the same size, but the predicted size varies.

The results are nevertheless promising in that a network trained purely with synthetic tree is able to produce a comprehensible result with real imagery. Adding domain adaptation is a logical next step, and we are considering how to create labeled datasets of real imagery.

Fig. 8: Real image of two apple trees with the output of the network.

VI. Conclusion and Future work

This paper has investigated the application of PV2BEV methods to agricultural environments. From a range of existing PV2BEV methods [6], three methods [7] [8] [9] were selected and assessed on orchard scenes. The PYVA algorithm was chosen for training with synthetic tree imagery, and testing on synthetic imagery for ground truth, and testing on real images for qualitative results.

Fig. 9: Real image of an apple orchard with 7 recognizable trees and the network output.

Fig. 10: Real image of an apple orchard containing 4 recognizable trees with the output of the network.

The results indicate that PV2BEV can be applied to trees. One next step is a fuller investigation of real imagery, while a key topic for future work is how to match the generated bird's eye view to real aerial imagery, which opens up several research possibilities. The initial investigation has set a promising basis for this future work.

Acknowledgments

Our thanks to Peter Frohlich of AgriCircle for allowing farm access for data collection, and SpeedTree for access to tree models.

References

[1] [Online]. Available: http://floatingrobotics.com
[2] [Online]. Available: http://ageagle.com
[3] [Online]. Available: http://rowesys.ethz.com
[4] [Online]. Available: http://ecorobotix.com
[5] "Under-canopy uav laser scanning for accurate forest field measurements," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 164, pp. 41–60, 2020.
[6] Y. Ma, W. Tai, X. Bai, H. Yang, Y. Hou, Y. Wang, Y. Qiao, R. Yang, D. Manocha, and X. Zhu, "Vision-centric bev perception: A survey," 2022.
[7] K. Mani, S. Daga, S. Garg, S. S. Narasimhan, M. Krishna, and K. M. Jatavallabhula, "Monolayout: Amodal scene layout from a single image," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 1689–1697.
[8] S.-Y. Yang, W.-C. Lee, Y.-C. F. Wang, and Y.-K. Lai, "Projecting your view attentively: Monocular road scene layout estimation via cross-view semantic alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 14 777–14 786.
[9] R. Zhang, H. Qiu, T. Wang, X. Xu, Z. Guo, Y. Qiao, P. Gao, and H. Li, "Monodetr: Depth-aware transformer for monocular 3d object detection," *arXiv preprint arXiv:2203.13310*, 2022.
[10] [Online]. Available: https://store.speedtree.com/speedtree-store/
[11] [Online]. Available: https://www.swisstopo.admin.ch/fr/geodata/height/alti3d.html