Efficient 3D Reconstruction of Multiple Plants from UAV Images with Deep Learning

Hong Huang
School of Computer Science and Technology, Guangdong University of Technology
Guangzhou, Guangdong, China
2112105227@mail2.gdut.edu.cn

Zhuowei Wang*
School of Computer Science and Technology, Guangdong University of Technology
Guangzhou, Guangdong, China
wangzhuowei0710@163.com

Genping Zhao
School of Computer Science and Technology, Guangdong University of Technology
Guangzhou, Guangdong, China
genping.zhao@gdut.edu.cn

ABSTRACT

Acquiring the 3D structure of plants is a critical task in the agricultural industry. Existing methods of generating 3D point clouds for multiple plants require a long processing time. In this paper, a 3D reconstruction method for numerous plants is proposed. Firstly, camera parameters in different viewpoints are obtained from the aerial image of plants by incremental structure from motion. Subsequently, the learning-based multi-view stereo takes images and the corresponding camera parameters as inputs to acquire initial depth maps. Finally, the depth maps are filtered and fused to produce a complete and dense 3D point cloud. We conducted experiments on an agricultural orchard dataset to compare with other methods. Experimental results demonstrate that our method reconstructs point clouds of plants with good quality while having a lower running time.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**.

KEYWORDS

3D reconstruction, Plant, Point cloud, UAV images, Deep learning

ACM Reference Format:
Hong Huang, Zhuowei Wang*, and Genping Zhao. 2024. Efficient 3D Reconstruction of Multiple Plants from UAV Images with Deep Learning. In *2024 16th International Conference on Machine Learning and Computing (ICMLC 2024), February 02–05, 2024, Shenzhen, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3651671.3651732

1 INTRODUCTION

Agriculture is one of the basic industries of mankind and it is of great importance for the economic development of the country. The three-dimensional (3D) structure of plants is a subject of great interest in agricultural activities. On the one hand, high-precision 3D models of plants help to understand and analyze the morphological characteristics of plants [14]. On the other hand, 3D measurements of vegetation can greatly facilitate the calculation accuracy of indicators such as vegetation cover and forest biomass [18]. Therefore, how to efficiently and accurately obtain 3D models of plants is an essential topic.

In the beginning, the 3D information of plants was captured by manual 3D digitization [17]. However, this method is laborious and time-consuming. The advance of digital agriculture allows greater 3D techniques under certain conditions. For example, in an indoor experimental setting, the 3D characterization of plants can be well captured, but the flexibility of such a method is low. Due to the gradual increase in agricultural activities, it is challenging to satisfy the needs of agricultural practitioners by only learning the spatial structure of a single plant. With the help of unmanned aerial vehicles (UAVs) and remote sensing, 3D modeling of multiple plants in a large scene becomes possible [24]. However, it takes a long time to reconstruct a complete 3D model because of high computational complexity.

The powerful capabilities of computer vision and deep learning have fueled the development of 3D technology. One of these technologies is image-based 3D reconstruction, which can recover 3D structures from 2D images [6]. There are three main types of 3D representations of objects: voxel, point cloud, and mesh [15]. Particularly, generating point clouds through depth maps is well suited for deep learning, and thus received widespread attention [23]. This approach is more advantageous in terms of computational time, which is very suitable for the 3D reconstruction of multiple plants. To the best of our knowledge, few studies can efficiently generate a multi-plant 3D point cloud with deep learning.

In light of the above information, we develop a deep learning-based 3D reconstruction method for plants in large numbers. In the first step, the algorithm obtains camera intrinsic and extrinsic parameters from different views using an incremental structure from motion technique applied to aerial images of plants. After that, we employ a learning-based multi-view stereo approach that utilizes the features of images and the camera to generate an initial depth map for each image. To produce a complete and dense 3D point cloud, the depth maps are finally filtered and fused.

2 RELATED WORKS

2.1 Traditional methods for 3D plant reconstruction

Conventionally, Structure from Motion (SFM) and Multi-View Stereo (MVS) are the two most used techniques for collecting 3D point clouds of plants from images [3]. Lin et al. [9] developed a platform

ICMLC 2024, February 02–05, 2024, Shenzhen, China

ACM ISBN 979-8-4007-0923-4/24/02
https://doi.org/10.1145/3651671.3651732

that can automatically gather the point cloud of a plant. Peng et al. [16] presented an SFM method for 3D plant reconstruction. Using this technique, a point cloud can be created from binocular vision images. However, the studies above only apply in conditions with a single plant present.

To better produce 3D models of multiple plants in one go, researchers have proposed more alternative methods. Photogrammetry is a technique suitable for large-scale 3D reconstruction of plants [7]. Zhang et al. [27] performed 3D reconstruction on a forest by UAV aerial photogrammetry. Liu et al. [11] used commercial software based on the SFM-MVS algorithm to generate a dense point cloud of a crop plot. Marks et al. [12] proposed a method to generate accurate 3D plant models from UAV imagery. It combines bundle adjustment and template matching to overcome the challenge of self-occlusion. Although the above methods have been successful in large-scale 3D plant reconstruction, they require relatively long processing times.

2.2 Deep learning-based methods for 3D plant reconstruction

With the advancement of computer vision, a few studies started focusing on deep learning-based 3D point cloud reconstruction of plants [8]. Zhao et al. [28] proposed a deep neural network named P3ES-Net for generating a 3D point cloud of a plant. However, this method is based on a single RGB image, which is inappropriate for multiple plants in an agricultural field. Chen et al. [4] reconstructed 3D point clouds of three different plants from multi-view images using MVSNet [25], which sped up the whole process. Due to the high memory consumption of MVSNet, this method is challenging to handle high-resolution images, so it can not be used for 3D plant reconstruction from UAV images.

3 METHODOLOGY

In this section, we describe the detailed pipeline of our method for 3D plant reconstruction. The general flow of the proposed technique is displayed in Figure 1.

3.1 Incremental structure from motion

SFM algorithms are categorized as incremental, global, and hierarchical. Compared with the other two categories, incremental SFM is more accurate and robust to outliers. Depth maps cannot be acquired without accurate camera parameters. Therefore, we choose to use this algorithm to obtain the spatial coordinates of the points in the 2D image, which benefits the accurate estimation of camera parameters. In our implementation, this is achieved by an incremental SFM module from [19].

3.2 Learning-based multi-view stereo

To obtain a denser point cloud, the outputs of the previous step are processed by this deep learning network. Initial depth maps are obtained for different viewpoints.

We use a three-layer feature pyramid network (FPN) [10] for feature extraction. The network comprises 10 layers with convolutional operations and 5 output layers. Given a dataset with N images of $H \times W$ resolution, when feature extraction is performed on an image, this image is treated as a reference image denoted as I_0. All images other than this one are treated as source images, denoted as $\{I_i\}_{i=1}^{N-1}$. Pixel-level features are extracted at multi-resolution for all the input images. In a course-to-fine way, the speed of estimating depth maps is accelerated. Since the image features are acquired hierarchically, the image features at various resolutions ($\frac{1}{2}H \times \frac{1}{2}W$, $\frac{1}{4}H \times \frac{1}{4}W$, $\frac{1}{8}H \times \frac{1}{8}W$) will be processed in a multi-stage way.

Next, the initial depth map is estimated for each reference image and its corresponding source images by the multi-stage Patchmatch module. The traditional Patchmatch algorithm is a randomized approach used for structural image editing [2]. This algorithm quickly finds approximate nearest neighbor field (NNF) matches between patches of images, through initialization, cost propagation, random search, and other iterative steps to quickly find matching pixel patches. Developed from traditional Patchmatch, learnable Patchmatch [22] estimates the depth map more efficiently. The core structure of the learnable Patchmatch in the multi-stage Patchmatch module is shown in Figure 2.

The learnable Patchmatch algorithm consists of five significant steps: initialization, adaptive propagation, matching cost computation, adaptive spatial cost aggregation, and depth regression. During the initialization, based on the preset depth range $[d_{min}, d_{max}]$, we sample each pixel with D_f depths in the inverse depth range that is equivalent to an image space with uniform sampling. We also divided D_f intervals in the inverse depth range to ensure the range is evenly covered. During the adaptive propagation, since the depth assumption value of the neighboring pixel may be better than the current pixel depth hypotheses, we need to consider the depth from the neighboring pixel. Propagating the depth of the neighbor pixel to the current pixel also serves as the hypothesis of the current pixel, which makes the estimated depth value more accurate. The implementation of the adaptive propagation is based on a 2D deformable convolution network (DCN), which takes the reference feature map as input, and calculates the additional offset of the coordinates of neighboring pixels on the same surface from the current pixel. It is defined as Equation 1:

$$D_p(p) = \{D(p + o_i + \Delta o_i(p))\}_{i=1}^{N_p} \tag{1}$$

where $D_p(p)$ are the depth hypotheses, D is the depth map from last iteration, o_i are fixed 2D offsets, $\Delta o_i(p)$ are additional 2D offsets, and N_p stands for the number of neighboring pixels. In matching cost computation, the source image feature map is warped under the depth layer corresponding to the reference image by using differentiable warping, and the actual pixels corresponding to pixel p on the source feature map are computed by using the camera parameters $\{K, R, T\}$ of the reference view and the source view, and then the source feature map after warping is obtained, and then the matching cost is computed as Equation 2:

$$p_{i,j} = K_i \cdot (R_{0,i} \cdot (K_0^{-1} \cdot p \cdot d_j) + t_{0,i}) \tag{2}$$

where $p_{i,j}$ is the j-th warped pixel in the i-th source map and d_j is the depth hypothesis. Considering the GPU memory consumption problem, we use the group-wise correlation to compute the cost of each depth hypothesis. The g-th group similarity is computed as Equation 3:

$$S_i(p, j)^g = \frac{G}{C} \left\langle F_0(p)^g, F_i(p_{i,j})^g \right\rangle \tag{3}$$

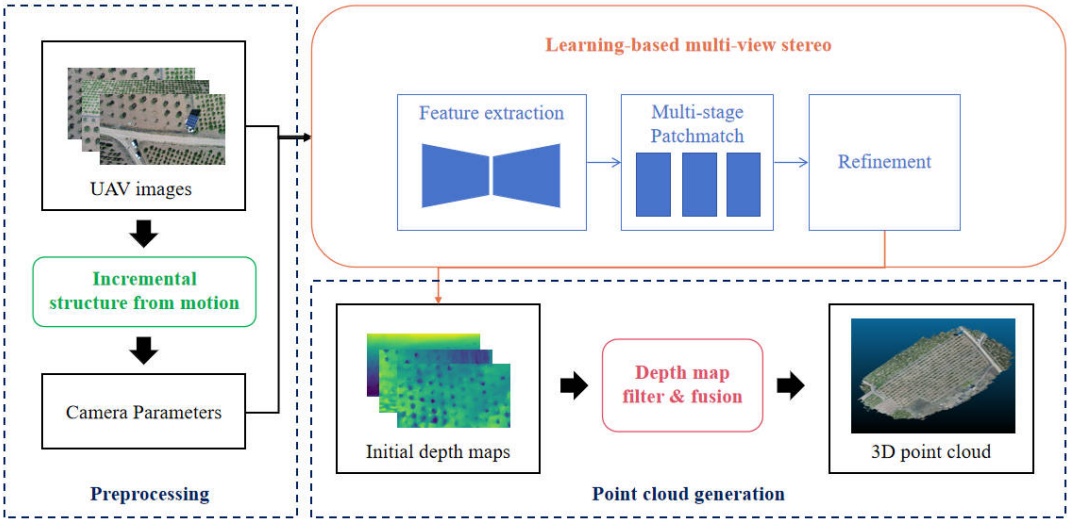

Figure 1: The overall pipeline of our method.

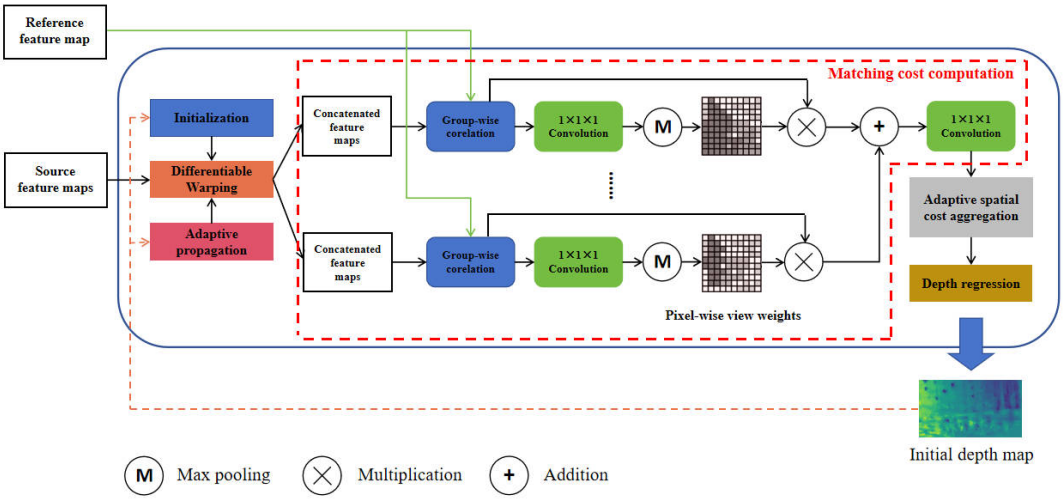

Figure 2: The structure of learnable Patchmatch in the multi-stage Patchmatch module.

where C is the number of feature channels, $\langle \cdot, \cdot \rangle$ is the inner product, $F_0(p)$ represents the features of the reference map, and $F_i(p_{i,j})$ represents the warped features of the source maps. The $N-1$ views are aggregated using the pixel view weights, an initial similarity set is used, and the sigmoid function is utilized to output a number between 0 and 1 corresponding to each pixel and calculate its weighted average. It is defined as Equation 4:

$$\overline{S}(p,j) = \frac{\sum_{i=1}^{N-1} w_i(p) \cdot S_i(p,j)}{\sum_{i=1}^{N-1} w_i(p)} \tag{4}$$

where $w_i(p)$ is the weight of pixel p, $S_i(p,j)$ represents per group similarity, and $\overline{S}(p,j)$ are the weighted sum of $S_i(p,j)$. In adaptive spatial cost aggregation, a 3D CNN network is used to compute the feature similarity between the neighboring pixels and the central pixel, which is defined as Equation 5:

$$\overline{C}(p,j) = \frac{\sum_{k=1}^{K_e} w_k d_k C(p + p_k + \Delta p_k, j)}{\sum_{k=1}^{K_e} w_k d_k} \tag{5}$$

where d_k are the weights of depth similarity, p_k represents the fixed pixel offset, and Δp_k stands for additional offset. At last, we use softmax to transform the cost volume into a probability volume and compute the expectation to achieve deep regression based on the probability distribution of each pixel. It is defined as Equation 6:

$$D(p) = \sum_{j=0}^{N-1} d_j \cdot P(p,j) \tag{6}$$

where $D(p)$ refers to the final depth, N is the number of the depth hypotheses at pixel p, d_j is the depth of j-th depth hypothesis, and $P(p, j)$ is the probability of pixel p.

The depth boundaries could be over-smoothed when sampling from a low-resolution depth map. The reference picture can be utilized to direct and optimize the depth map output by Patchmatch because it comprises boundary information in the input reference image. As a result, we employ a depth residual network that learns and outputs residuals, adds the residual values to the depth map output from Patchmatch, and then produces an optimized depth map.

Similar to other deep learning-based MVS networks, we use a loss function $Loss_{total}$ to compute the loss between the estimated depth map and the ground truth at each stage as well as the refined depth map loss. The loss function is defined as Equation 7:

$$Loss_{total} = \lambda \sum_{k=1}^{3} \sum_{i=1}^{n_k} Loss_i^k + (1-\lambda) Loss_{ref}^0 \tag{7}$$

where $0 < \lambda < 1$, $Loss_i^k$ represents the loss of iteration i on stage k ($k = 1, 2, 3$) and $Loss_{ref}^0$ represents the loss of refinement. Both $Loss_i^k$ and $Loss_{ref}^0$ adopt smooth L1 loss.

3.3 Depth map filtering and fusion

Since the initial depth maps generated by the previous step will have some outliers, they need to be filtered. Two indicators from [26]: photometric and geometric consistencies are employed to achieve better filtration results. To maintain a high quality of matching, pixels with a threshold below 0.8 in each depth map will be considered outliers. Pixels with a reprojection depth error lower than 0.01 are treated as outliers as well. The depth values corresponding to these pixels will be removed in our experiments. After filtering the depth maps, we adopt the method from [13] for fusion, which minimizes depth violations and occlusions from various angles. All filtered depth maps are fused to obtain a unified 3D point cloud. The point cloud fused by our method will be shown in the next section.

4 EXPERIMENT

We performed point cloud reconstruction experiments on an agricultural dataset to evaluate the effectiveness of our method. Additionally, We compared the quality and time consumption of reconstruction with other methods.

4.1 Dataset and experimental settings

Our study employed an aerial image dataset of a large orchard [21]. The RGB images in the dataset are divided into three categories based on various shooting angles: Nadir, Oblique, and Mix. The Mix set is the combination of the Nadir set and the Oblique set. In addition, a 3D point cloud model of the entire orchard is also offered for accessing the reconstruction outcomes. Examples of images and the point cloud are shown in Figure 3 and Figure 4, respectively. Table 1 provides additional details about the orchard dataset. All the experiments were conducted on a high-performance platform with a 24 GB NVIDIA RTX A5000 GPU and an AMD EPYC 7543 CPU.

4.2 Evaluation metrics

To evaluate the reconstruction quality of the point cloud, we employed three metrics that are commonly used in 3D reconstruction [1]:

- **Accuracy(Acc.)**: the average distance between the reconstructed point cloud and the ground-truth point cloud, which is defined as Equation 8:

$$Accuracy(R, G) = \frac{1}{|R|} \sum_{r \in R} \min_{g \in G} \|r - g\| \tag{8}$$

  where R stands for the reconstructed point set, G for the ground-truth point set, r for a reconstructed point from R, and g for a ground-truth point from G.
- **Completeness(Comp.)**: the average distance between the ground-truth point cloud and the reconstructed point cloud, which is defined as Equation 9:

$$Completeness(R, G) = \frac{1}{|G|} \sum_{g \in G} \min_{r \in R} \|g - r\| \tag{9}$$

  where R, G, r, and g are the same as in Equation 8.
- **Overall(OA.)**: the average of the first two metrics, which is defined as Equation 10:

$$Overall(R, G) = \frac{1}{2}[Accuracy(R, G) + Completeness(R, G)] \tag{10}$$

Additionally, we logged the point cloud reconstruction time in seconds, which was used to compare the speeds of different methods.

4.3 Experimental results

We evaluate the proposed method on three types of images (Nadir, Oblique, and Mix) from the dataset. The methods used for comparison were Colmap [20], PIE-UAV and CasMVSNet [5]. Colmap is an end-to-end image-based 3D reconstruction pipeline that utilizes the SFM-MVS algorithm. PIE-UAV (Beijing PIESAT Information Technology Co., Ltd., Beijing, China) is a professional UAV image processing software, and it can produce a complete 3D point cloud from aerial images. CasMVSNet is a deep learning-based network that can be used for 3D reconstruction from multi-view images.

Figure 5 presents the results of the point clouds reconstructed by the four methods. It can be found that the point clouds reconstructed by our method were denser than those of other methods. Furthermore, Table 2 summarizes the quality evaluation of the point cloud. The metrics in the table were measured in meters, the smaller the value the better. As can be seen in the table, PIE-UAV is the best in all three metrics of the Nadir images, while our method is the second best. In the other two sets of images (Oblique and Mix), our method performed best for Completeness and Overall metrics. The reason for the above results is related to how the images were taken. Nadir images are taken vertically, while oblique images are taken at a certain angle, and the latter can better capture the 3D structure of the object than the former. In our method, we use an FPN to learn the features of the images, which can make better use of the information in oblique and mixed images than in nadir images.

The running time of the four methods is shown in Table 3, which is measured in seconds. According to the table, the time needed

Figure 3: Example images of the orchard dataset.

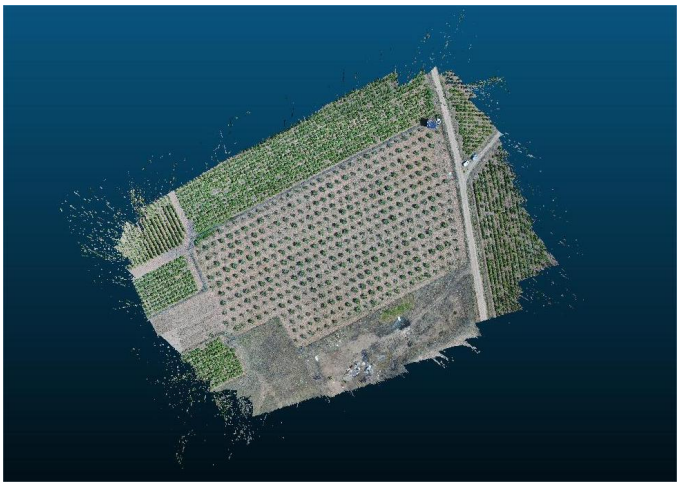

Figure 4: The ground-truth point cloud of the orchard.

Table 1: Details of the agricultural dataset.

Dataset	Category of Images	Numer of Images	Shooting Angle	Image Resolution
Orchard	Nadir	86	0°	5475 × 3078
	Oblique	162	30°	5475 × 3078
	Mix	248	0° and 30°	5475 × 3078

Table 2: Evaluation metrics of point cloud quality. The data in the table is in meters, the lower the value the better.

Method	Nadir			Oblique			Mix		
	Acc.	Comp.	OA.	Acc.	Comp.	OA.	Acc.	Comp.	OA.
Colmap	0.613	0.532	0.573	0.140	0.118	0.129	**0.144**	0.121	0.132
PIE-UAV	**0.138**	**0.136**	**0.137**	**0.137**	0.126	0.131	**0.144**	0.114	0.129
CasMVSNet	0.523	0.472	0.498	0.142	0.144	0.143	0.146	0.139	0.143
Ours	0.437	0.480	0.458	0.147	**0.063**	**0.105**	0.174	**0.073**	**0.123**

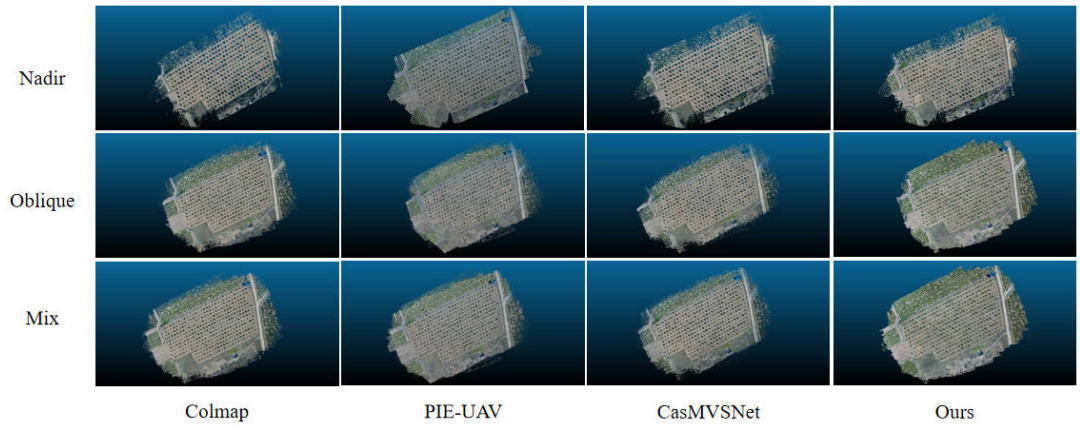

Figure 5: Comparison of point clouds with other methods.

Table 3: Comparison of running time. All data in the table are in seconds, the lower the value the better.

Method	Nadir	Oblique	Mix
Colmap	226.98	443.28	731.04
PIE-UAV	437.88	706.94	995.43
CasMVSNet	204.88	380.17	706.76
Ours	**168.08**	**366.88**	**637.98**

for reconstruction grew as the number of images did. Our method took 168.08, 366.88, and 664.57 seconds to reconstruct point clouds, which was the best of all methods. Colmap and PIE-UAV apply traditional multi-view stereo in their pipelines. Compared with the previous two methods, our method employs a learning-based multi-view stereo strategy for initial depth map generation, resulting in a shorter running time to generate the point cloud. In addition, our approach still has a benefit over CasMVSNet, another deep learning-based method. The reason for this is that CasMVSNet builds 3D cost volumes with 3D convolutional neural networks, which is time-consuming. Our network relies on the Patchmatch algorithm rather than the concept of keeping a structured cost volume at all. With the help of the adaptive spatial cost aggregation in the multi-stage Patchmatch module, the intrinsic spatial coherence of depth maps is used to discover a better result quickly without having to consider all possibilities, thus accelerating the inference of the depth map.

5 CONCLUSION

In this study, a fast and accurate method for multi-plant 3D reconstruction based on deep learning is presented. By combining conventional SFM with the learnable MVS module, our method reduces the time to generate depth maps, allowing for faster acquisition of the 3D point cloud of plants. According to the experimental results, our approach efficiently reconstructs point clouds of plants while maintaining high quality. Although our method has the benefit of fast reconstruction speed, it does not perform well on a small number of images or photos taken vertically. Our future research direction is how to reach better 3D reconstruction of plants using fewer images.

ACKNOWLEDGMENTS

This work was sponsored in part by Provincial Agricultural Science and Technology Innovation and Extension Project of Guangdong Province under Grant 2023KJ147, and in part by Guangzhou Science and Technology Plan Project under Grant 202201011835.

REFERENCES

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. 2016. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision* 120 (2016), 153–168.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* 28, 3 (2009), 24.

[3] EF Berra and MV Peppa. 2020. Advances and challenges of UAV SFM MVS photogrammetry and remote sensing: Short review. In *2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS)*. IEEE, 533–538.

[4] Zhen Chen, Hui Lv, Lu Lou, and John H Doonan. 2022. Fast and accurate 3D reconstruction of plants using mvsnet and multi-view images. In *Advances in Computational Intelligence Systems: Contributions Presented at the 20th UK Workshop on Computational Intelligence, September 8-10, 2021, Aberystwyth, Wales, UK 20*. Springer, 390–399.

[5] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. 2020. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2495–2504.

[6] Xian-Feng Han, Hamid Laga, and Mohammed Bennamoun. 2019. Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE transactions on pattern analysis and machine intelligence* 43, 5 (2019), 1578–1604.

[7] Jakob Iglhaut, Carlos Cabo, Stefano Puliti, Livia Piermattei, James O'Connor, and Jacqueline Rosette. 2019. Structure from motion photogrammetry in forestry: A review. *Current Forestry Reports* 5 (2019), 155–168.

[8] San Jiang, Wanshou Jiang, and Lizhe Wang. 2021. Unmanned aerial vehicle-based photogrammetric 3d mapping: A survey of techniques, applications, and challenges. *IEEE Geoscience and Remote Sensing Magazine* 10, 2 (2021), 135–171.

[9] Chenhui Lin, Hong Wang, Chengliang Liu, and Liang Gong. 2020. 3D reconstruction based plant-monitoring and plant-phenotyping platform. In *2020 3rd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*. IEEE, 522–526.

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125.

[11] Fusang Liu, Pengcheng Hu, Bangyou Zheng, Tao Duan, Binglin Zhu, and Yan Guo. 2021. A field-based high-throughput method for acquiring canopy architecture using unmanned aerial vehicle images. *Agricultural and Forest Meteorology* 296 (2021), 108231.

[12] Elias Marks, Federico Magistri, and Cyrill Stachniss. 2022. Precise 3D reconstruction of plants from UAV imagery combining bundle adjustment and template matching. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2259–2265.

[13] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. 2007. Real-time visibility-based fusion of depth maps. In *2007 IEEE 11th International Conference on Computer Vision*. Ieee, 1–8.

[14] Seishi Ninomiya. 2022. High-throughput field crop phenotyping: current status and challenges. *Breeding Science* 72, 1 (2022), 3–18.

[15] Niall O'Mahony, Sean Campbell, Lenka Krpalkova, Daniel Riordan, Joseph Walsh, Aidan Murphy, and Conor Ryan. 2019. Computer Vision for 3D Perception: A Review. In *Intelligent Systems and Applications: Proceedings of the 2018 Intelligent Systems Conference (IntelliSys) Volume 2*. Springer, 788–804.

[16] Yeping Peng, Mingbin Yang, Genping Zhao, and Guangzhong Cao. 2021. Binocular-vision-based structure from motion for 3-D reconstruction of plants. *IEEE Geoscience and Remote Sensing Letters* 19 (2021), 1–5.

[17] Jessada Phattaralerphong and Hervé Sinoquet. 2005. A method for 3D reconstruction of tree crown volume from photographs: assessment with 3D-digitized plants. *Tree Physiology* 25, 10 (2005), 1229–1242.

[18] Worasit Sangjan, Rebecca J McGee, and Sindhuja Sankaran. 2022. Optimization of UAV-based imaging and image processing orthomosaic and point cloud approaches for estimating biomass in a forage crop. *Remote Sensing* 14, 10 (2022), 2396.

[19] Johannes L Schonberger and Jan-Michael Frahm. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4104–4113.

[20] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. 2016. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 501–518.

[21] Sergio Vélez, Rubén Vacas, Hugo Martín, David Ruano-Rosa, and Sara Álvarez. 2022. High-Resolution UAV RGB Imagery Dataset for Precision Agriculture and 3D Photogrammetric Reconstruction Captured over a Pistachio Orchard (Pistacia vera L.) in Spain. *Data* 7, 11 (2022), 157.

[22] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. 2021. Patchmatchnet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14194–14203.

[23] Xiang Wang, Chen Wang, Bing Liu, Xiaoqing Zhou, Liang Zhang, Jin Zheng, and Xiao Bai. 2021. Multi-view stereo in the deep learning era: A comprehensive review. *Displays* 70 (2021), 102102.

[24] Huang Yao, Rongjun Qin, and Xiaoyu Chen. 2019. Unmanned aerial vehicle for remote sensing applications—A review. *Remote Sensing* 11, 12 (2019), 1443.

[25] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. 2018. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*. 767–783.

[26] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. 2019. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5525–5534.

[27] Yanchao Zhang, Hanxuan Wu, and Wen Yang. 2019. Forests growth monitoring based on tree canopy 3D reconstruction using UAV aerial photogrammetry. *Forests* 10, 12 (2019), 1052.

[28] Genping Zhao, Weitao Cai, Zhuowei Wang, Heng Wu, Yeping Peng, and Lianglun Cheng. 2022. Phenotypic parameters estimation of plants using deep learning-based 3-D reconstruction from single RGB image. *IEEE Geoscience and Remote Sensing Letters* 19 (2022), 1–5.