Title

MMM and CLCFM: A 3D point cloud reconstruction method based on photogrammetry for all-around images taken by rotating an individual plant

Authors

Atsushi Hayashi[1,2], Nobuo Kochi[1, 2, 3*], Kunihiro Kodama[1†], Sachiko Isobe[1†], Takanari Tanabata[1*]

Affiliations

[1] Kazusa DNA Research Institute, Chiba, 292-0818, Japan.

[2] National Agriculture and Food Research Organization, Ibaraki 305-8518, Japan.

[3] R&D Initiative, Chuo University, Tokyo, 112-8551, Japan.

[†] Present address: The University of Tokyo, Tokyo, 188-0002, Japan.

[*]Address correspondence to: imageapplabo@kochi.email.ne.jp and tanabata@kazusa.or.jp

Abstract

This research proposes a novel technique for acquiring a large amount of high-density, high-precision 3D point cloud data for plants. We propose two methods, multi-masked matching (MMM) and the closed-loop coarse-to-fine method (CLCFM). The proposed approach addresses challenges in reconstructing plant 3D point clouds from all-around images using SfM and multi-view stereo methods. Given the complex structure of plants, with thin objects like leaves and stems overlapping, reconstructing accurate 3D point clouds presents significant challenges. One of these is occlusion, where points in the 3D point cloud cannot be obtained due to overlapping parts, preventing accurate point capture. Another is the generation of erroneous points in non-existent locations due to image matching errors along object outlines. The MMM mitigates these problems through an iterative process that alternates between locally generating point clouds and refining these point clouds by removing erroneous points using mask images. The proposed pipeline enables the construction of plant 3D point clouds with minimal noise and fewer missing points. By facilitating the acquisition of high-density, high-precision 3D data on a large number of plant bodies, as required for research activities, the proposed method is expected to enable comparative analysis of visible phenotypes in the growth process of a wide range of plant species based on 3D information.

MAIN TEXT

1. Introduction

Shape data related to the morphology and structure of individual plants and organs, as well as the changes in these parameters throughout the plant growth cycle, provide crucial information for clarifying plant functions. With advances in molecular biology, the acquisition of large volumes of detailed morphological data has become increasingly important. Digital imaging technology offers significant advantages over conventional measurement methods, particularly in its capacity for non-destructive measurement. The development of digital cameras and related technologies has further increased the use of imaging techniques due to their low cost and ease of operation [1].

However, several challenges arise when measuring the size, shape and structure of objects using images captured by a camera. Since a camera image is necessarily a two-dimensional projection, accurately obtaining three-dimensional (3D) shape and structure information, including depth, from a single image is inherently difficult. Therefore, to achieve a more detailed and comprehensive understanding of visible phenotypes during the plant growth process, it is necessary to acquire accurate information about plant morphology and structure from sources other than single photographs. This requires precise measurement and digitization of the shape of plant organs and the overall structure of the plant body in the same 3D space as real space. To address this issue, significant progress is being made in the development and application of measurement technology to obtain 3D information about plants [2-4].

Various methods exist for obtaining 3D information about an object, including active techniques that use lasers or other light sources to illuminate the object for measurement, and passive techniques that rely on camera images for measurement. We focused on structure from motion (SfM) and multi-view stereo (MVS) techniques, two passive photogrammetric methods for acquiring 3D information [5]. This approach was chosen because it derives measurements simply from images captured with a digital camera, offering a low-cost and efficient solution for obtaining accurate data [6].

Photogrammetry is a process that uses digital cameras to take multiple images of an object from various angles, and then uses these images to reconstruct the object's 3D information as 3D point cloud data. This process uses two different types of processing: SfM processing and MVS processing. SfM processing aims to arrange all the images of an object taken from various angles in 3D space, based on the position from which they were taken in relation to the object [7, 8]. Using the objects or parts of objects in the images as landmarks, the images are compared to determine how they appear, and from the results, the position in 3D space (hereafter, camera position) at which each image was taken in relation to the object is estimated. The MVS process searches for the location of points on the object in each image, and uses the results of this search and the camera position information to determine the location of the points in 3D space using stereo matching. The process is repeated to calculate the location of the points on the object in 3D space, and a dense 3D point cloud of the object in the image is reconstructed [9].

Therefore, accurately estimating the camera position in the SfM process and precisely identifying matching points in the MVS process are essential for obtaining high-quality 3D point cloud data of the target object. Several methods have been proposed to address

these challenges. For improving the accuracy of camera position estimation in SfM processing, several approaches have been suggested. One method involves placing 3D reference points at strategic locations around the object to be measured, which are then used to estimate the position of the camera [10, 11]; another method leverages location information obtained from GPS to minimize estimation errors [12]. Additionally, for SfM processing involving images captured in a manner that encircles the object, a loop-closing technique has been proposed to minimize the camera position estimation error by linking the first and last images in the sequence [13]. To reduce the accumulation of matching errors caused by the detection of similar patterns during the image matching process used for camera position estimation, another approach has been proposed that involves pre-calculating the overall image similarity and the relationships between images [14-16].

Various methods have been proposed for the 3D reconstruction of plants [17-21]. Plants present unique challenges for 3D reconstruction due to their characteristic shapes, such as thin stems and leaves, and the similar textures of these surfaces. These features make plants particularly susceptible to errors in the image matching process used in SfM and MVS processing when comparing multiple images. Consequently, when plants are used as measurement targets, the resulting 3D point clouds often suffer from issues such as noise [17] or missing points [18]. Several methods have been proposed to solve these issues, such as optimizing the lighting used during filming [19], adding post-processing to remove noise points[20], and adding post-processing to complete in missing areas [21].

The aim of this research was to develop technology for acquiring large quantities of high-density, high-precision 3D point cloud data for plants. To accomplish this goal, we first proposed a method for solving the issues involved in SfM/MVS processing for plants by using images taken of all surroundings of the target plant to reconstruct 3D point clouds with less noise and fewer missing points. We then developed the pipeline scripts for implementing this method. Finally, we conducted validation experiments and showed that the proposed method and the pipeline enable the acquisition of 3D point clouds of plants with less noise and point loss. By satisfying the requirement for acquisition of large amounts of high-density, high-precision 3D point cloud data of plants for research activities, our proposed approach is expected to allow comparative analyses of visible phenotypes in the growth process of a wide range of plant species based on 3D information.

2. Materials and Methods

Plant material and cultivation

A total of four varieties were used: three soybean varieties, GmJMC092 (KUROHIRA), GmJMC112 (FUKUYUTAKA) and HOJAK, were derived from the Genebank project, NARO (https://www.gene.affrc.go.jp), and another one soybean varieties, MISUZUDAIZU, wase derived from the NBRP Lotus japonicus & Glycine max (https://nbrp.jp/en/resource/lotus-glycine-en/). The seeds were sown in 1/5000a Wagner pots in 2018 using baked soil, and the plants were grown in a greenhouse at the Kazusa DNA Research Institute. Water tubes were installed in the pots, and irrigation was carried out twice a day.

3D reconstruction algorithm

The proposed 3D reconstruction method is based on all-around images of the plant to be measured, which are taken by rotating it in front of the camera, and uses SfM and MVS to

obtain 3D point clouds from the images. When acquiring the all-around images, the information on the order of the images is recorded in the image files themselves, and this information is used to execute local MVS repeatedly. This local processing solves the problems of noise and missing points in the 3D point cloud reconstruction of plants.

MMM: Multi-Masked Matching

The 3D point cloud generation method we propose, MMM, involves generating a 3D point cloud by performing stereo matching using images captured from closely positioned cameras. This is followed by iterative point removal using mask images derived from the surrounding viewpoint images utilized during point cloud generation.

Figure 1 shows a representative example of the process. In this example, stereo matching is performed using two images from adjacent camera positions (indicated by the pink squares in Fig. 1A, Process 1) to generate a 3D point cloud, referred to as the *baseCloud*. This process is executed sequentially using the Metashape API functions buildDepthMaps () and buildDenseCloud ().

Next, a mask image is created for the images used to acquire the *baseCloud* (the 72 pink squares in Fig. 1A, Process 2), which were taken from viewpoints within a 45-degree range to the left and right, and the process of removing points from the *baseCloud* is executed. The mask image is a binary image in which the area of the plant region to be measured in the image taken is 1 and the other areas are 0. This process is carried out by executing the Metashape API functions selectMaskedPoints () and removeSelectedPoints () in that order. Figure 1B shows the algorithm for constructing a 3D point cloud using MMM; in this example, the images are taken by setting up four cameras above and below the target and rotating the target 5 degrees at a time. For each image taken, the process is carried out for all combinations of adjacent images to obtain the point cloud. Finally, all the point clouds obtained by this process are integrated to form a 3D point cloud of the target plant.

CLCFM: Closed-Loop Coarse-to-Fine Method for SfM

We propose a CLCFM for improving the stability and accuracy of camera position estimation for images taken from all around the target. Instead of performing SfM using all images taken from around the target at once, we perform SfM using a small subset of the images taken from the target, and then repeat SfM by adding the subset of the remaining images taken from the target. By repeating SfM adding a small number of all-around images in this way, and then finally using all the images to perform SfM, the camera positions of all the images taken are estimated. This method of performing SfM sequentially rather than all at once improves the accuracy of the camera position estimation.

The principle of the method is illustrated using an example where a single camera captures images of an object that is rotated in 5-degree increments (Fig. 2A). First, the images are divided into some groups. In this example, the 72 images taken from around the object are divided into six groups. Next, the order in which the SfM process is executed (GroupOrder) is determined. The GroupOrder is selected so that the bias in the estimated camera positions is minimized. For example, if Group 1 is the first group subjected to the SfM process, then Group 4, which includes the images halfway between the adjacent images of Group 1, is selected as the second group to be processed, and Group 2, which is between Group 1 and Group 4, is selected as the third.

Next, the procedure shown in Fig. 2B is followed to perform SfM sequentially for each group. The images in Group 1 are used for the first camera position estimation. The main process for camera position estimation in Fig. 2B, AlingProcess (), is executed for ImgList [] in the order of the Metashape API functions matchPhotos (), alignCameras (), and optimizeCameras (). Then, the results of the first camera position estimation are used as initial values, and a second SfM is performed by adding the images of Group 4 to the images of Group 1. Similarly, images are processed in the order specified by GroupOrder, with SfM being sequentially executed using the camera position estimates obtained from the previous step as initial values. The final camera position estimates are derived by processing all groups. When images are captured using multiple cameras arranged vertically, those taken at the same vertical angle are classified into the same group and processed accordingly.

3D point cloud reconstruction pipeline program

We developed a 3D point cloud reconstruction pipeline that implements our proposed method. This pipeline consists of four subprograms that are executed sequentially with the captured image as the initial input to reconstruct the 3D point cloud (Fig. 3).

1) CreateMask.cpp

This is a program for creating mask images. The mask images output by this program are binary images in which the areas of the plants that reconstruct the 3D point cloud and the areas of the objects necessary for this image processing are white, and the rest are black. Mask images are produced for all captured images and are used for mask processing and noise removal in the CLCFM and MMM methods.

This program was developed using Visual C++ (Visual studio 2019) and OpenCV version 3.4.14.

2) align_camera.py

This script performs the camera position estimation process of the CLCFM algorithm. The camera position estimation information obtained by this process is used in MMM. This program was developed using Python 3.8, Metashape Professional Edition 1.8.5, and the Python 3 Module (Agisoft, https://www.agisoft.com).

3) build_densecloud.py

This script executes the MMM algorithm to construct a 3D point cloud and outputs multiple 3D point cloud data files in PLY format. The script was developed using Python 3.8, Metashape Professional Edition 1.8.5, and the Python 3 Module. The results presented in this paper were obtained using the following execution parameters for the processing functions within the Python 3 Module.

- matchPhotos(): The images were processed at 100% scale with accuracy set to 'high'.
- buildDepthMaps(): The images were processed at 1/4 scale (1500x1000 pixels) with quality set to 'medium'.

4) merge_subsampling.py

This program integrates the multiple 3D point cloud data output by 'build_densecloud.py' and performs subsampling processing at intervals of 0.5 mm. This process outputs the 3D point cloud of the target object.

This program was developed using Python 3.7 and open3D 0.15.

Imaging

In order to obtain the best images for the proposed methods MMM and CLCFM, we built an imaging studio consisting of a rotating stage with plants, a camera, LED lighting, and

background panels (Fig. 4A). The background panels, equipment, and cords were colored blue for generating the mask images used in MMM.

The imaging stage was designed so that the plant could be rotated in front of the camera to obtain a perspective image of the entire plant (Fig. 4B). SGSP-160YAW (Sigma Koki, Tokyo) and GIP-101 (Sigma Koki) were connected to a personal computer using a serial USB converter (COM-1(USB)H; Contec, Tokyo), and the rotations were controlled by a program running on the personal computer.

To hold the cultivation pot, an acrylic disc (35 cm in diameter) was made and attached to the SGSP-160YAW. Small pieces of paper printed with coded targets were attached to the acrylic disc, and additional coded targets were attached to the three pillars fixed outside the perimeter of the disc. The coded targets were used as reference points for integrating the point cloud using MMM, and to define the scale that gives the actual dimensions [5]. The coded targets were attached to the following positions: the center of the disc; eight positions at 45-degree intervals on the circumference 15 cm from the center; and positions 20 cm apart vertically on the pillars. Each of the three pillars consisted of an aluminum frames of 2.5 cm width × 1.5 cm thickness ×1 m length, to which was attached a strip of paper printed with a random dot image. These random dots were used to ensure that the feature points obtained in the CLCFM and MMM processing steps were stable [5]. The pillars were attached to the side of the disc at equal angles (Fig. 4B, left).

Images were captured using four digital SLR cameras (D5500; Nikon Corporation) and lenses (AF-S Micro NIKKOR 60mm f/2.8G ED, Nikon Corporation). Figure 4C shows the layout of the cameras used to capture images of the plants used as targets (plant height 400–1200 mm). The camera was connected to a personal computer via a USB cable. The shooting conditions were: shutter speed 1/15 sec, ISO sensitivity 100, aperture F/16. The acquired image was saved as a 6000x4000 pixel JPG file.

3. Results

'*baseCloud*' construction process in MMM

Figure 5 shows the results of an experiment conducted to verify the optimal viewing angle for generating *baseCloud* point clouds of the target plant. Figures 5a and 5b display portions of the *baseCloud* reconstructed using only two adjacent images and four adjacent images, respectively, from different camera positions. The images were captured by rotating the table holding the object in 5-degree increments, resulting in viewing angles of 5 degrees for the *baseCloud* in Fig. 5a and 15 degrees for that in Fig. 5b. No significant difference was observed between the *baseClouds* in Figs. 5a and 5b for this object. However, when reconstructing another sample using Metashape 1.6, we confirmed that setting the viewing angle to 5 degrees helped reduce the loss of point clouds (Fig. S3). The processing time required for reconstruction using MMM decreases as the number of images used decreases. Based on these findings, we chose to generate *baseCloud* using two adjacent images.

Viewing angle of the mask image used in the point removal process in MMM

We next investigated the optimal viewing angle for the mask image used in the point removal process of *baseCloud* performed by MMM. To determine the optimal angle, we performed the point removal process of *baseCloud* by MMM under four different viewing angles of the mask image in the same way as shown in Fig. 6. Figure 6 shows the results when the viewing angle of the mask image used was a) 50 degrees, b) 90 degrees, c) 180 degrees, and d) 360 degrees. The results indicated that the more the viewing angle of the surrounding mask was increased, the more points were removed, but points in the plant

body that did not need to be removed were also deleted. We therefore decided to set the viewing angle of the mask image used in the point removal process at 90 degrees.

3D point cloud reconstruction of soybean plants

Using the pipeline developed and the processing conditions established in the previous sections, we successfully reconstructed 3D point clouds for four different soybean varieties (Fig. 7a, Fig. S1). The results demonstrated the ability of the proposed method to reconstruct 3D point clouds of varying shapes, sizes, and structural complexities. For comparison, the same image sets were processed using standard Metashape functions to generate 3D point clouds. This comparison confirmed that the proposed method effectively mitigated the appearance of blue dots as noise in the outlines of leaves and stems, in addition to addressing the problem of point clouds failing to form in certain thin stem regions (Fig. 7b, Fig. S1, Fig. S2).

Figure 7c shows the result of not performing the point removal process after generating the *baseCloud* with MMM, and Fig. 7d shows the result of performing the point removal process after generating the *baseCloud* with MMM. In both cases, point cloud reconstruction and point removal processing were performed after SfM was executed using CLCFM. Without point removal processing, noise points were generated in the background and other parts of the captured image (Fig. 7c), while point removal processing resulted in the removal of noise points (Fig. 7d).

Effects of the point removal processes in MMM

We compared the differences in the reconstruction of 3D point clouds by the MMM with and without point removal processing. *Base clouds* were generated using the MMM point cloud reconstruction method, followed by the output of 3D point clouds both after applying point removal processing (Fig. 8a) and without it (Fig. 8b). These two 3D point clouds were then overlaid and compared within a virtual space. To facilitate this comparison, only the plant body was extracted, while the support, pot, and rotating stage were manually removed. The distance between corresponding points in the aligned 3D point clouds was calculated, and points with a distance of 3 mm or less were deleted, yielding a point cloud that represented the differences between the two 3D reconstructions (Fig. 8c). This process was repeated for distance ranges of 0–1 mm, 1–2 mm, 2–3 mm, and 3–22 mm, and the number of points constituting each point cloud was recorded (Fig. 8d). The analysis revealed that 78,637 points had a distance difference of 3 mm or more in the point cloud after point removal processing, accounting for 4.3% of the total points in the point cloud (Fig. 8d).

4. Discussion

This paper proposed two algorithms, MMM and CLCFM, for the 3D reconstruction of images of a plant taken from all directions. Since plants have a shape with thin objects such as leaves and stems overlapping, when performing 3D point cloud reconstruction using SfM/MVS, there are issues with points in the 3D point cloud not being obtained due to occlusion caused by overlapping parts, and points being generated in places where the object does not exist due to image matching errors in the outline of the object.

To solve this problem, we proposed the MMM. The MMM is a method that repeatedly executes a process for reconstructing point clouds locally (multi matching) and a process for removing point clouds using a mask image from the generated point clouds (multi mask matching) (Fig. 1). To execute MMM, it is a prerequisite that the camera position estimation result is obtained with high accuracy by SfM processing. To meet this requirement, we proposed CLCFM, which estimates the camera positions with high

accuracy for images that capture the entire surroundings of the target. When SfM is executed using images that capture the entire surroundings of the target, there is a problem of accumulated estimation errors. CLCFM selects the images for executing SfM processing and executes SfM repeatedly to suppress the accumulation of cumulative errors (Fig. 2). MMM and CLCFM are based on the assumption that the target is rotated and photographed from all sides, and that the order of the images is known. Using this information, we were able to implement the proposed method by executing local point cloud reconstruction and selecting mask images for point removal processing.

We built a studio for taking images (Fig. 4) and an image analysis pipeline (Fig. 3) to implement the proposed method, and then performed 3D point cloud reconstruction for soybean plants (Fig. 7). We confirmed that it is possible to reconstruct a high-density 3D point cloud of a plant body that suppresses the loss of thin stems, which are a characteristic shape of the plant body, and suppresses the noise of 3D points that occur in places where plant body do not exist (Fig. 8).

In this paper, Metashape Professional Edition 1.8 was used to execute SfM/MVS. We found that the different versions of Metashape yielded different results of MVS point cloud reconstruction (Fig. S3, Fig. S4). However, regardless of the version, the method proposed in this paper is effective for reconstructing 3D point clouds of plant bodies with overlapping organ shapes such as leaves and stems, as it can suppress noise by performing point cloud reconstruction using local views with MMM and point removal processing using mask images.

Generation of local 3D points using images with a narrow angle of view and removal of noise points using mask images

MMM is a method that avoids occlusion by performing 3D point cloud reconstruction using images with a narrow angle of view. However, when searching for corresponding points between images using a narrow angle of view for 3D point cloud construction, if the corresponding points are on the contour, a mismatch problem arises. This mismatch increases the number of points that become noise (points that are generated in places where they should not exist). To solve this problem, we augmented the MMM method with a newly developed algorithm for removing erroneously generated points using mask images.

Figure 9 shows a schematic diagram of the image-matching process at the boundary of plant organs.

To estimate the 3D position of point P on the contour line of the object, we obtain the coordinates of point P_{A1} and point P_{A2}, which indicate the position of point P in the images taken by cameras A1 and A2. Using these coordinates and the camera position information for cameras A1 and A2 obtained by SfM, the depth distance DP between point P and the cameras is calculated, the position of point P in 3D space is determined based on this result, and a point for the 3D point cloud is generated (Fig. 9A).

Here, we focus on point P in the outline of the plant body. Image matching is performed to find position P_{A2} on the image taken by camera A2 relative to position P_{A1} on the image taken by camera A1.

The DP value changes according to the matching results. If the position of point P found by camera A1 is correctly matched with point P in the image from camera A2, the DP will be given the correct value, and as a result, point P on the plant body will be generated in 3D space (Fig. 9A). However, if the point P is mismatched and the matching result is different from the correct point P, the DP value will be different from DP and will acquire DP_m and $DP_{m'}$ values. Based on these values, points are generated at the positions Pm and Pm', which are different from the correct position of point P. These points P_m and $P_{m'}$ are

noise points that do not exist on the plant body. The causes of image matching not being detected correctly include the resolution of the image sensor of the camera used for imaging, and matching with points with similar textures to the leaves and stems in the vicinity of point P. For the problem of noise points caused by mismatching, we proposed a method that does not reduce mismatching, but instead finds many candidate points and adds a step for removing points in post-processing. Using this method, it becomes possible to reduce the number of noise points.

Figure 9C shows a schematic diagram of the point removal process for MMM. Cameras A1, A2, B1, and B2 represent the camera positions. The images captured by Cameras A1 and A2 are used to generate the point cloud (*baseCloud*), and the mask images generated from the images captured by Cameras A1, A2, B1, and B2 are used for the point removal process (Fig. 9C). The mismatching that occurs at points on the contour line of the object occurs in the vicinity of the intersection lines $P_{A1}P_{back}$ and $P_{A2}P'_{back}$ of Cameras A1 and A2. Using the mask images of Cameras A1 and A2, the point removal process can remove mismatching points at locations other than the object. However, none of the mismatch points generated on the $P_{A1}P_{back}$ and $P_{A2}P'_{back}$ boundary lines can be removed. The mismatching points P_m and $P_{m'}$ on the boundary line shown in Fig. 9C can be removed by removing the areas other than the object at the intersection lines $P_{B1}P_{b1}$ and $P_{B2}P_{b2}$ of Cameras B1 and B2 in the mask image. By removing the mismatch points on the line between $P_{A1}P_{back}$ and $P_{A2}P'_{back}$, it becomes possible to remove the mismatch points generated on the line between $P_{A1}P_{back}$ and $P_{A2}P'_{back}$.

The point cloud generated by MMM can achieve point cloud reconstruction with few noise points by performing stereo matching on a small number of images with a narrow viewing angle and removing noise points that occur outside the plant body using mask processing. However, the problem of generating points inside stems and leaves remains. When stereo matching is performed with a small number of images with a narrow field of view, many matching errors occur; the errors that occur outside the plant body in the contour area are deleted by mask processing, but the points inside the stem and leaves are not deleted and remain.

Improved accuracy of camera position estimation for all-around images with narrow inter-viewpoint angles

In this study, we used all-around images taken by rotating the object by 5 degrees for the process MMM. In SfM, there is a risk of camera position estimation failure due to the accumulation of camera position estimation errors for all-around images taken at short intervals (Fig. 10A). CLCFM performs the camera position estimation while gradually adding panoramic images. First, camera position estimation is performed using only images that surround the object but have an open viewpoint, and the results are used as initial values. Next, we add a few more images and run the camera position estimation process again, updating the results. We repeat this process, adding a few more images and running the estimation process until we have estimated the camera positions for all the images in the full surround. This processing method is expected to reduce the failure of position estimation due to error accumulation by suppressing the propagation of estimation errors that occur in the estimation process (Fig. 10B).

However, there is a problem with implementing this algorithm. If the distance between the cameras is long in the initial stage of processing, there are few common feature points between the images, and it becomes difficult to execute camera position estimation in the initial processing. Therefore, we solved this problem by using a support with a random pattern when taking pictures (Fig. 4B). It is possible to secure many image feature points in the location with the random pattern, and it is possible to execute bundle adjustment

with self-calibration [22]. As a result, it is possible to stably obtain the internal and external parameters of the camera. Finally, since it is not necessary to perform internal and external calibration in advance, there is also the advantage that it is possible to achieve high measurement accuracy without performing device setup, maintenance, or maintenance [5].

Acknowledgments

General: We would like to thank Keiko Saegusa, Rumi Fujihira, Satomi Nakayama, Chie Kasashima and Sayaka Kikuchi whose technical support at the Kazusa DNA Research Institute.We also thank to Prof. Ryo Akashi, Dr. Hidenori Tanaka, Dr. Masatsugu Hashiguchi and Dr. Takuyu Hashiguchi for providing soybean materials in this study.

Author contributions: HA, KN, IS, and TT contributed to the design of the study. HA made the 3D reconstruction pipeline programs. HA and TT made imaging hardware. HA, KN, and KK and conducted experiments and analyzed and interpreted data. HA, KN, IS, and TT wrote and revised the manuscript. KN and TT supervised the study.

Funding: This work was supported by a CREST Grant (No. JPMJCR16O1) from the Japan Science and Technology Agency and the Kazusa DNA Research Institute Foundation.

Competing interests: The authors declare that there is no conflict of interest regarding the publication of this article.

Data availability: The computer codes and scripts of MMM and CLCFM are deposited in a GitHub repository at https://github.com/tanasoft/MMM-CLCFM. Download links to image and 3D point cloud data that support the findings of this study are also provided at this GitHub repository.

Supplementary Materials

Figure S1 Results of reconstruction of 3D point clouds of four soybean varieties using the proposed method.

Figure S2 Results of reconstruction of 3D point clouds for the image in Fig. S1 using Metashape.

Figure S3 Results of running the analysis in Fig. 5 using Metashape 1.6.

Figure S4 Results of running the analysis in Fig. 6 using Metashape 1.6.

References

[1] Li Z, Guo R, Li M, Chen Y, Li G. A review of computer vision technologies for plant phenotyping. Computers and Electronics in Agriculture. 2020; 176:105672. https://doi.org/10.1016/j.compag.2020.105672

[2] Okura F. 3D modeling and reconstruction of plants and trees: A cross-cutting review across computer graphics, vision, and plant phenotyping. Breeding Science. 2022;72(1):31-47. https://doi.org/10.1270/jsbbs.21074

[3] Zhang Y, Teng P, Aono M, Shimizu Y, Hosoi F, Omasa K. 3D monitoring for plant growth parameters in field with a single camera by multi-view approach. Journal of Agricultural Meteorology. 2018;74(4):129-139. https://doi.org/10.2480/agrmet.D-18-00013

[4] Kunita I, Morita MT, Toda M, Higaki T. A Three-Dimensional Scanning System for Digital Archiving and Quantitative Evaluation of Arabidopsis Plant Architectures. Plant and Cell Physiology. 2021;62(12):1975-1982. https://doi.org/10.1093/pcp/pcab068

[5] Kochi N, Hayashi A, Shinohara Y, Tanabata T, Kodama K, Isobe S. All-around 3D plant modeling system using multiple images and its composition. Breeding Science. 2022;72(1):75-84. https://doi.org/10.1270/jsbbs.21068

[6] Salter WT, Shrestha A, Barbour MM. Open source 3D phenotyping of chickpea plant architecture across plant development. Plant Methods. 2021;17(1):95. https://doi.org/10.1186/s13007-021-00795-6

[7] Snavely N. Bundler: Structure from motion (sfm) for unordered image collections.2008. [accessed 30 Aug 2024] https://www.cs.cornell.edu/~snavely/bundler/

[8] Wu C. Towards Linear-Time Incremental Structure from Motion. Proceedings of the 2013 International Conference on 3D Vision. 127-134. https://doi.org/10.1109/3DV.2013.25

[9] Furukawa Y, Ponce J. Accurate, Dense, and Robust Multiview Stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010;32(8):1362-1376. https://doi.org/10.1109/TPAMI.2009.161

[10] Barazzetti L, Forlani G, Remondino F, Roncella R, Scaioni M. Experiences and achievements in automated image sequence orientation for close-range photogrammetric projects. Proceedings of Videometrics, Range Imaging, and Applications XI. 2011; 8085:80850F. https://doi.org/10.1117/12.890116

[11] Anai T, Kochi N, Yamada M, Sasaki T, Otani H, Sasaki D, Nishimura S, Kimoto K, Yasui N. Examination about influence for precision of 3D image measurement from the ground control point measurement and surface matching. Joint Workshop with ISPRS WG IV/7 and WG V/4. 2015; XL-4/W5:165-170. https://doi.org/10.5194/isprsarchives-XL-4-W5-165-2015

[12] Sattler T, Torii A, Sivic J, Pollefeys M, Taira H, Okutomi M,Pajdla T. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017:1637-1646.

[13] Scaramuzza D, Fraundorfer F, Pollefeys M. Closing the loop in appearance-guided omnidirectional visual odometry by using vocabulary trees. Robotics and Autonomous Systems. 2010;58(6):820-827. https://doi.org/10.1016/j.robot.2010.02.013

[14] Havlena M, Torii A, Pajdla T. Efficient Structure from Motion by Graph Optimization. Proceedings of the 11th European Conference on Computer Vision. 2010:110-113. https://doi.org/10.1007/978-3-642-15552-9_8

[15] Sweeney C, Sattler T, Höllerer T, Turk M, Pollefeys M. Optimizing the Viewing Graph for Structure-from-Motion. Proceedings of the 2015 IEEE International Conference on Computer Vision. 2015:801-809.

[16] Schönberger JL, Frahm JM. Structure-from-Motion Revisited. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. 2016:4104-4113. https://doi.org/10.1109/CVPR.2016.445

[17] Gibbs JA, Pound MP, French AP, Wells DM, Murchie EH, Pridmore TP. Active Vision and Surface Reconstruction for 3D Plant Shoot Modelling. IEEE/ACM

transactions on computational biology and bioinformatics. 2019;17(6):1907-1917. https://doi.org/10.1109/TCBB.2019.2896908

[18] Santos TT, Oliveira AA. Image-based 3D digitizing for plant architecture analysis and phenotyping. Proceedings of the Workshop on Industry Applications (WGARI) in SIBGRAPI 2012 (XXV Conference on Graphics, Patterns and Images). 2012. http://doi.org/10.13140/2.1.4576.1608

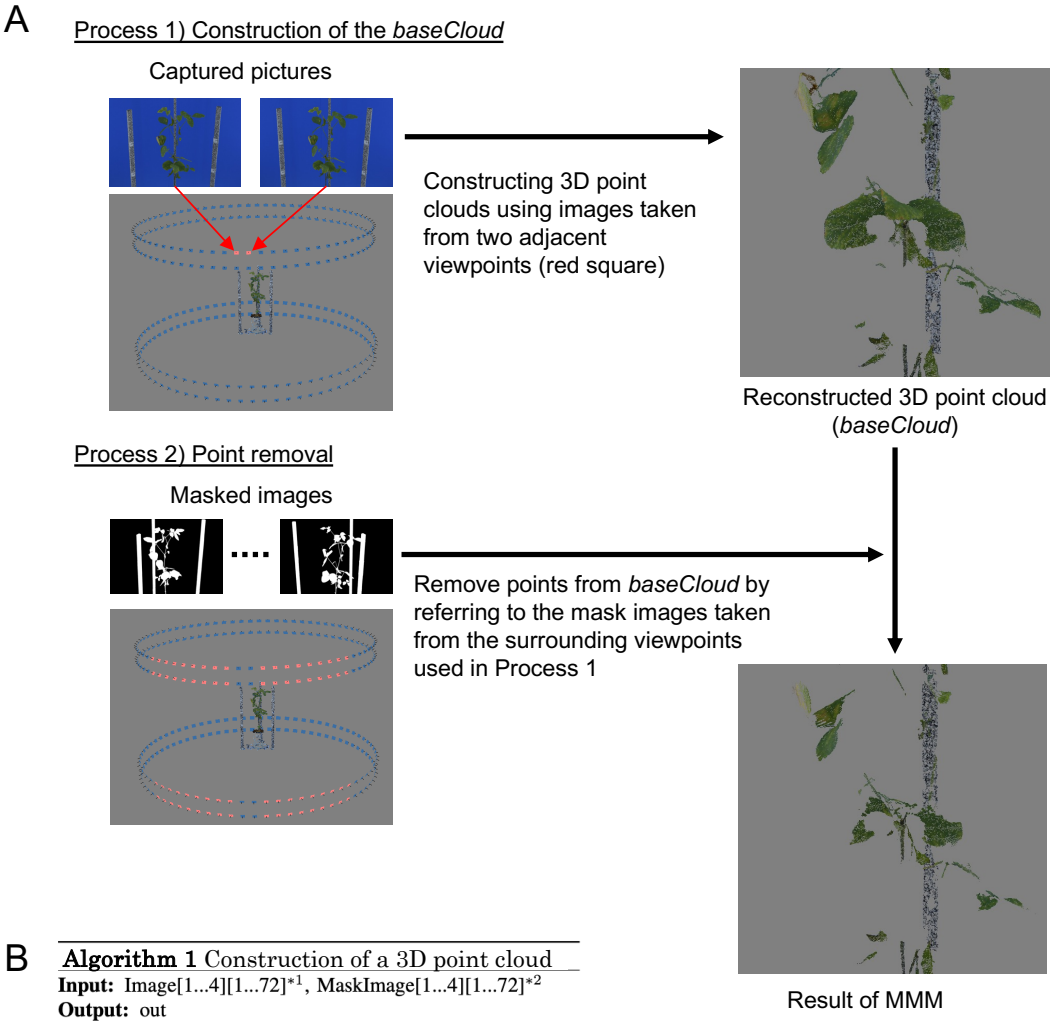
[19] Zhang Y, Teng P, Shimizu Y, Hosoi F, Omasa K. Estimating 3D leaf and stem shape of nursery paprika plants by a novel multi-camera photography system. Sensors. 2016;16(6):874. https://doi.org/10.3390/s16060874

[20] Li Y, Liu J, Zhang B, et al. Three-dimensional reconstruction and phenotype measurement of maize seedlings based on multi-view image sequences. Front Plant Sci 2022; 13: 974339. https://doi.org/10.3389/fpls.2022.974339

[21] Chen H, Liu S, Wang C, et al. Point cloud completion of plant leaves under occlusion conditions based on deep learning. Plant Phenomics 2023; 5: 0117. https://doi.org/10.34133/plantphenomics.0117

[22] Karara HM. Chap. 5 Camera-calibration in non-topographic photogrammetry. Maryland (USA). American Society for Photogrammetry and Remote Sensing; 1989.

A



B

Algorithm 1 Construction of a 3D point cloud

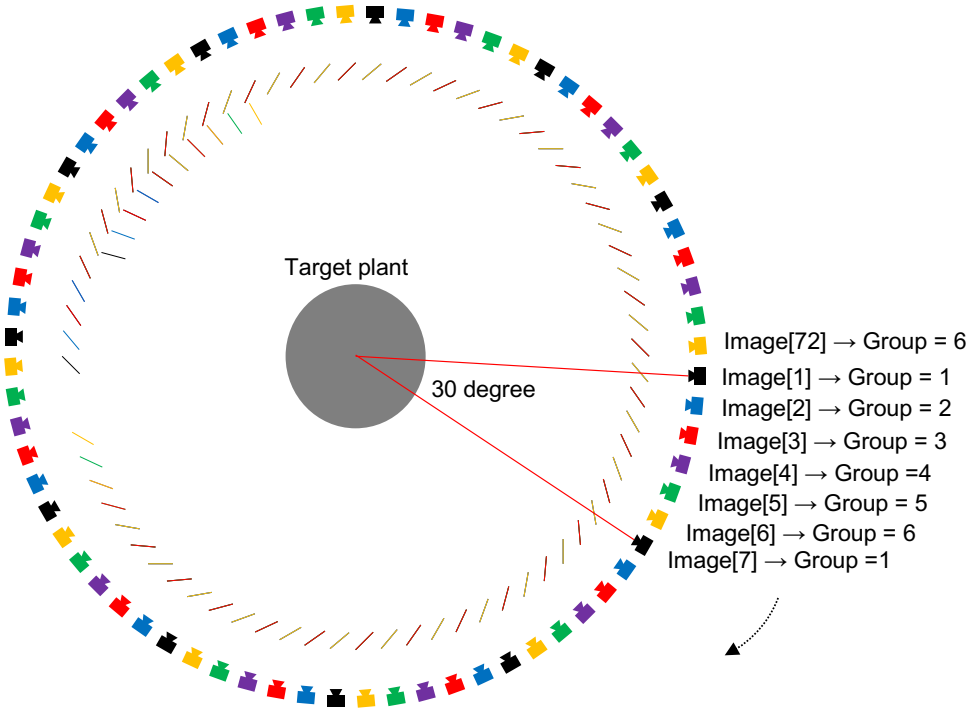
Input: Image[1...4][1...72][*1], MaskImage[1...4][1...72][*2]
Output: out

```
RemoveRange = 9
ResultPointCloud[] = {}
for i = 1 to 72 do
    CN1 = i
    if i = 72 then
        CN2 = 1
    else
        CN2 = i+1
    end if
    for c = 1 to 4 do
        baseCloud[]=Generate3DPointCloud(Image[c][CN1],
        Image[c][CN2])
        for i = 1 to RemoveRange do
            for c = 1 to 4 do
                RemovalProcess(baseCloud,MaskImage[c][CN1-
                i])
                RemovalProcess(baseCloud,MaskImage[c][CN2+i])
            end for
        end for
        ResultPointCloud[] += baseCloud[]
    end for
end for
return ResultPointCloud
```

[*1] Image[c][n]: captured image
[*2] MaskImage[c][n]: Masked image created from the captured image
c= camera order (1,2,3,4), n=captured sequence number(1,2,…,72)

Figure 1. *MMM:* Multi masked matching algorithm

A



Image[n] : Captured images, n=captured sequence number(1,2,…,72)

B

Algorithm 2 SfM by the closed-loop coarse-to-fine algorithm

Input: Image[1...72], GroupOrder[6]={1,4,2,5,3,6}
Output: out

```
ImgList[] = {}
AlignedData[] = {}
for i = 1 to 6 do
    for j = 1 to N do
        if (j mod 6 = GroupOrder[i]) then
            ImgList[] append Image[j]
        end if
    end for
    AlignedData[]=CameraAlignProcess(ImgList[],
    AlignedData[])
end for
return AlignedData[]
```

Figure 2. *CLCFM*: Closed-loop coarse to fine algorithm

1

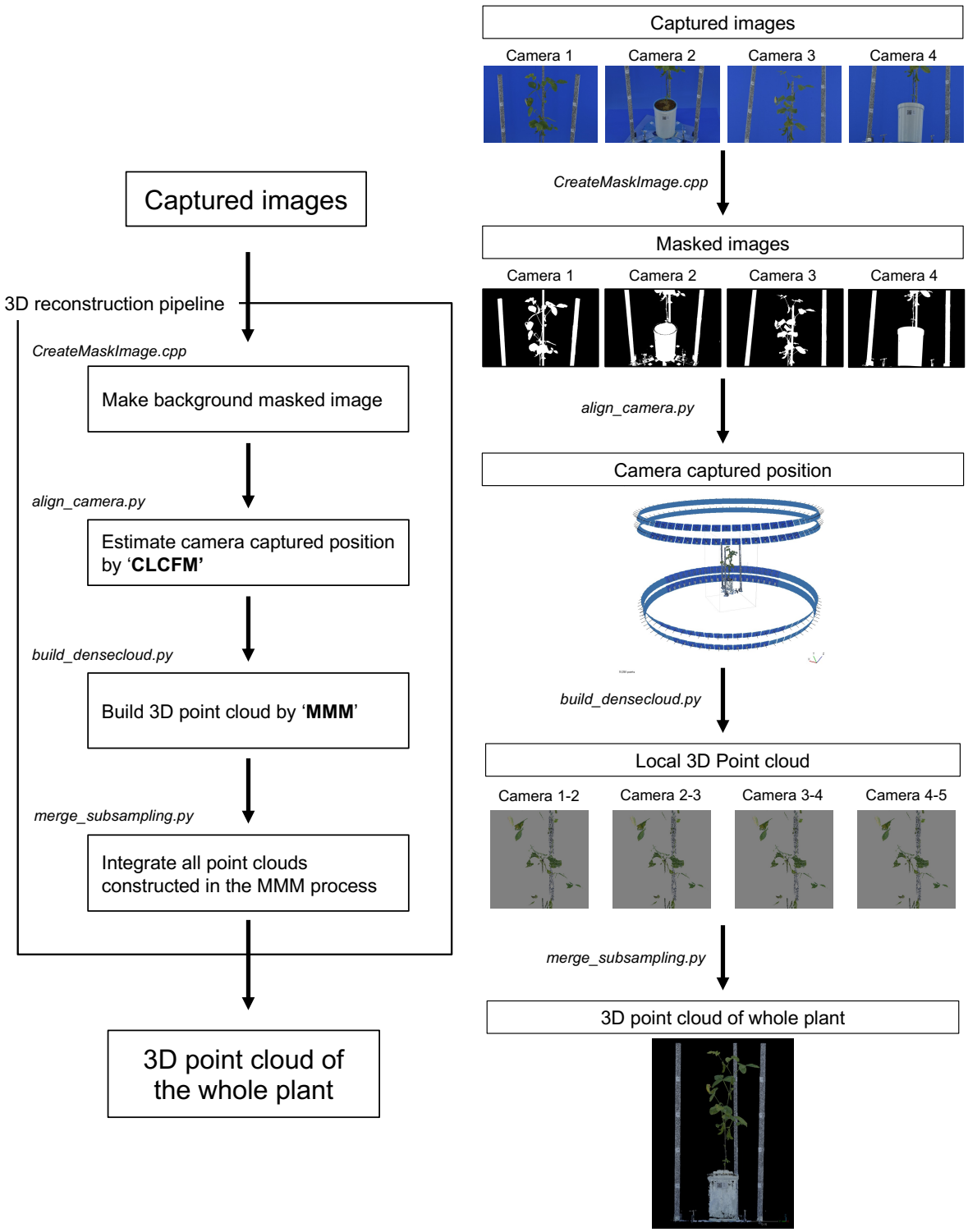


Figure 3. 3D point cloud reconstruction pipeline program

1

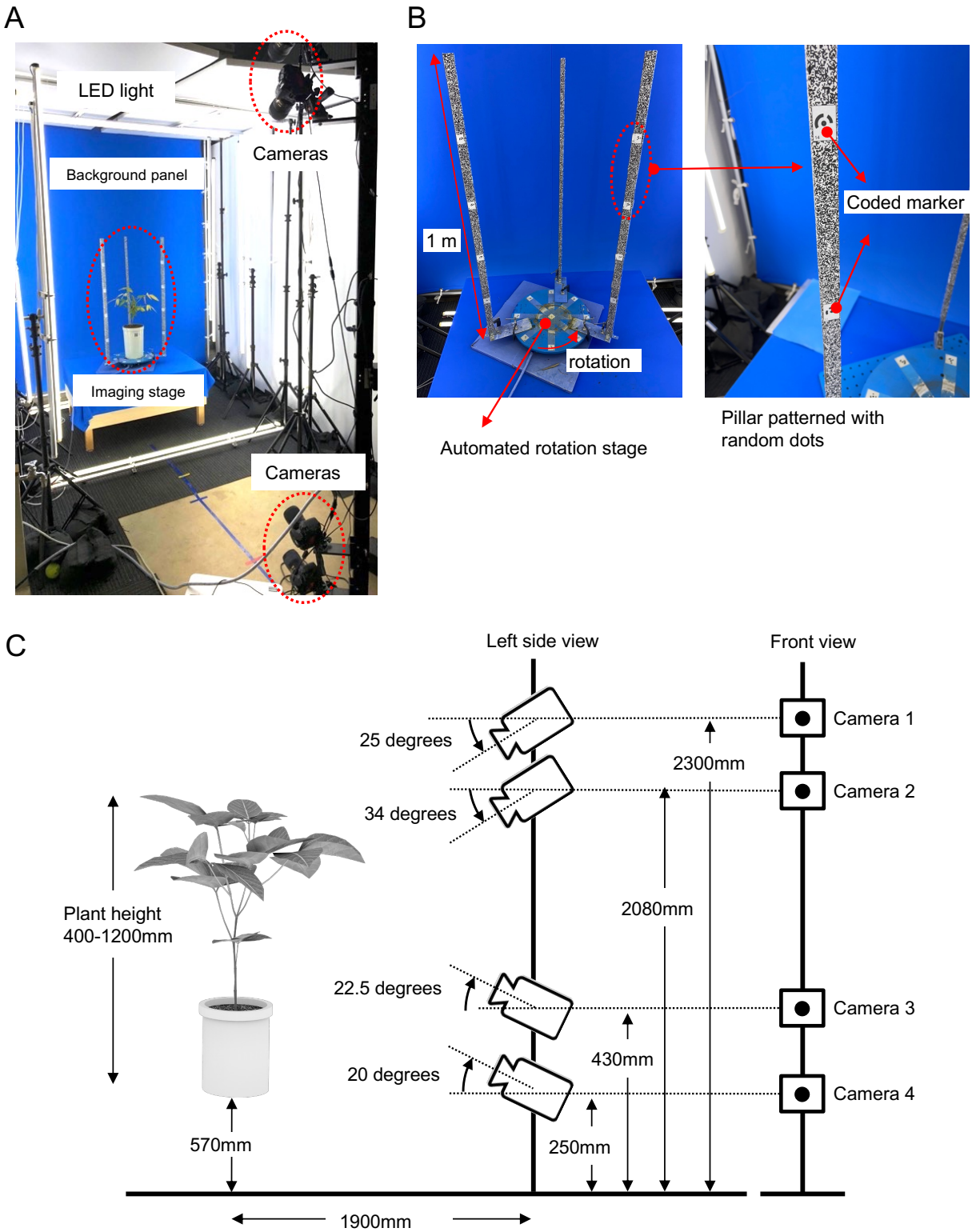
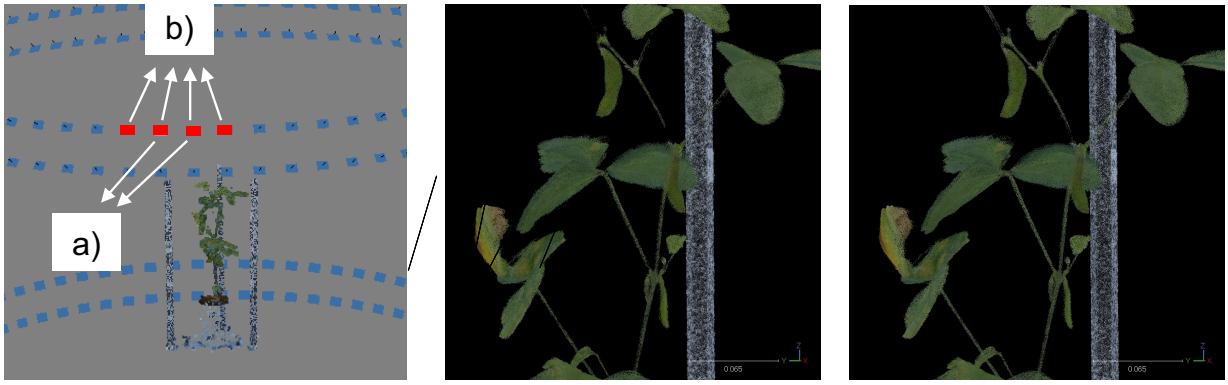


Figure 4. The imaging studio for 3D point cloud reconstruction by MMM and CLCFM

1



Camera position

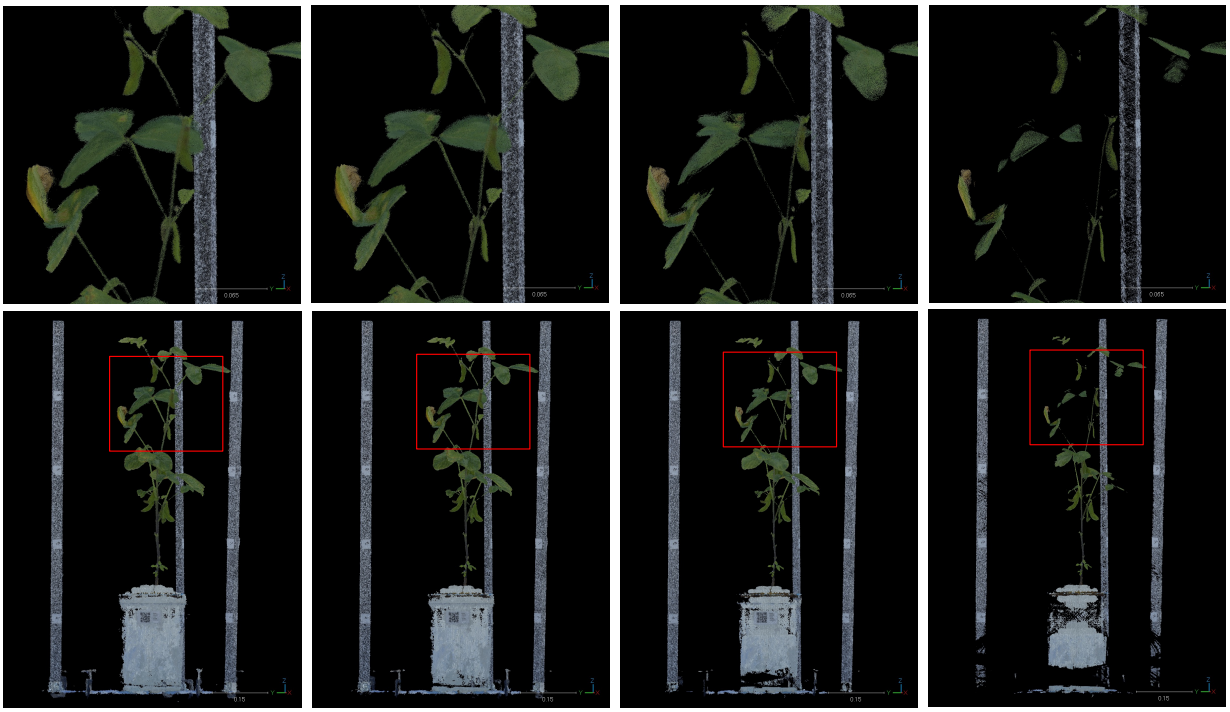
a) Executed MVS with two images

b) Executed MVS with four images

The plants in this result are GmJMC092 (KUROHIRA).

Figure 5. '*baseCloud*' construction process in MMM

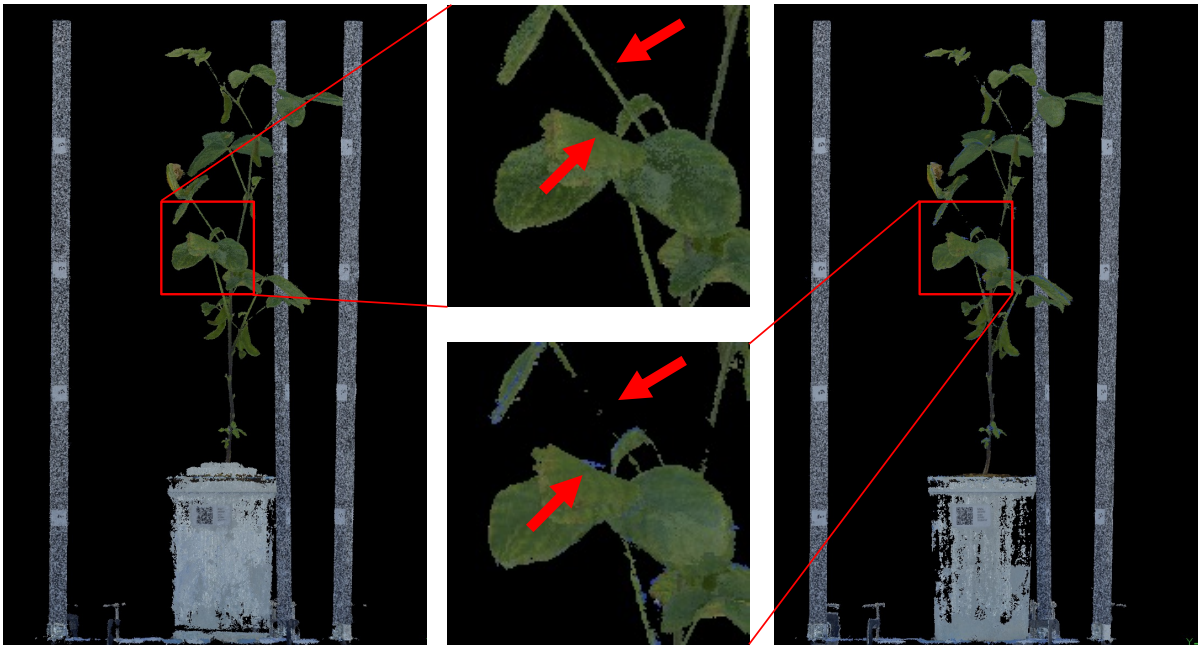
1



a) Viewing angle 50 degrees | b) Viewing angle 90 degrees | c) Viewing angle 180 degrees | d) Viewing angle 360 degrees

The plants in this result are GmJMC092 (KUROHIRA).

Figure 6. Results of point cloud reconstruction using different viewing angles for the mask image used in the point removal process



a) MMM processed

b) Standard Metashape processed

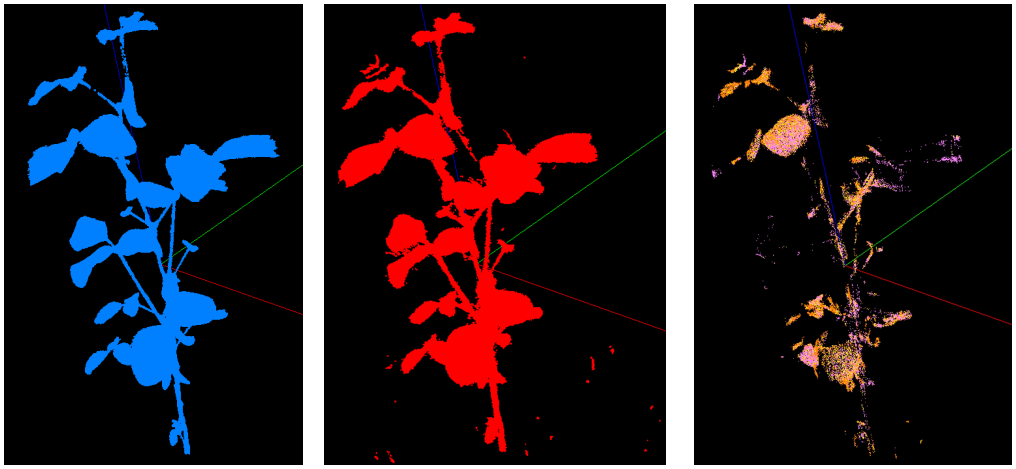
c) Without the point removal process

d) With the point removal process

The plants in this result are GmJMC092 (KUROHIRA).

Figure 7. 3D point cloud reconstruction of soybean plants

1



a) Point removal process performed (proposed method)

b) Point removal process NOT performed

c) Difference points from (a) and (b) more than 3mm

d) Comparison of the number of points

Range(mm)			number of points	%
0	~	3	1732066	95.7
3	~		78637	4.3
Total			1810703	100.0

Figure 8. Comparing the difference in point cloud reconstruction with and without the point removal process by mask images

1

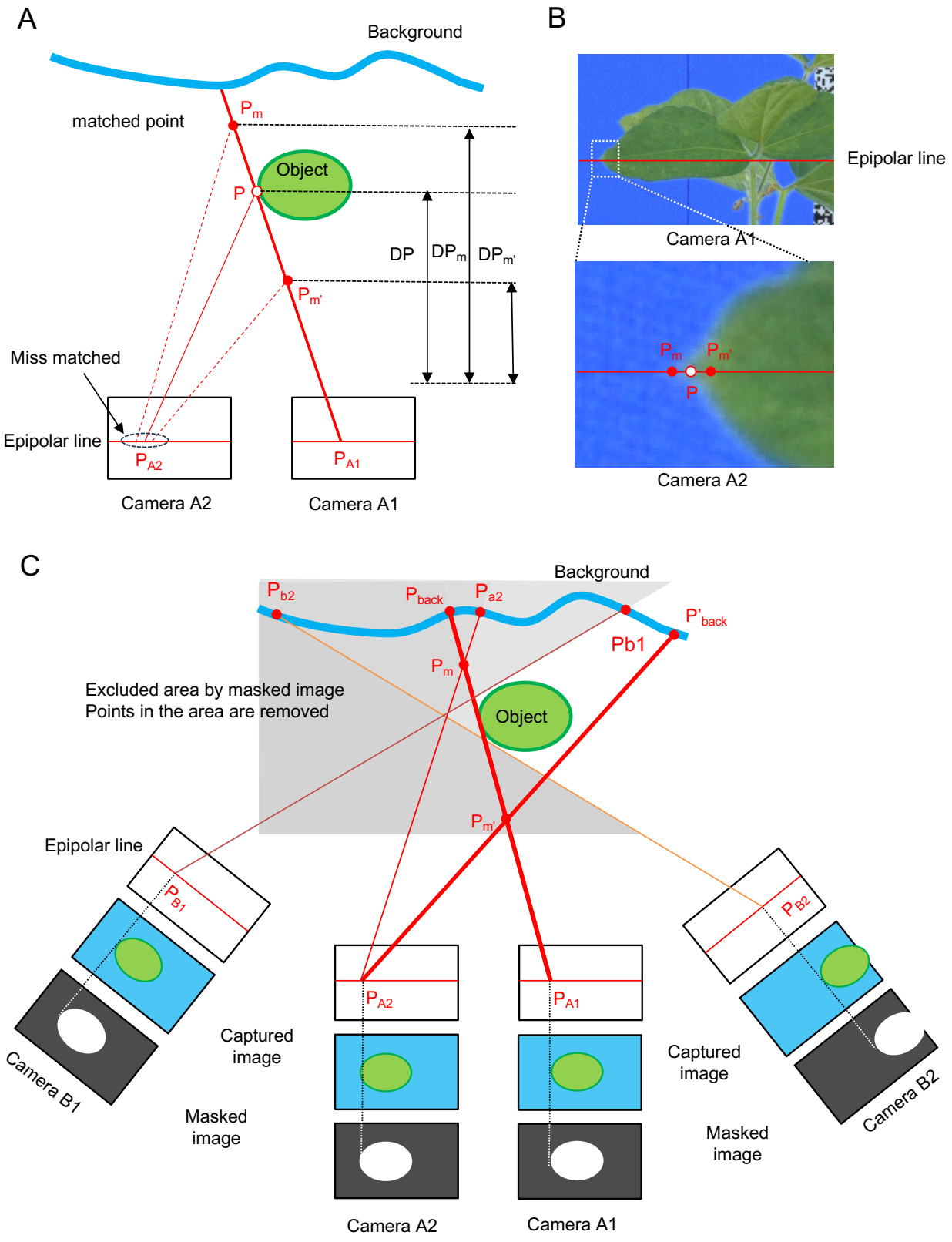


Figure 9. Schematic diagram of the image-matching process at the boundary of plant organs

1

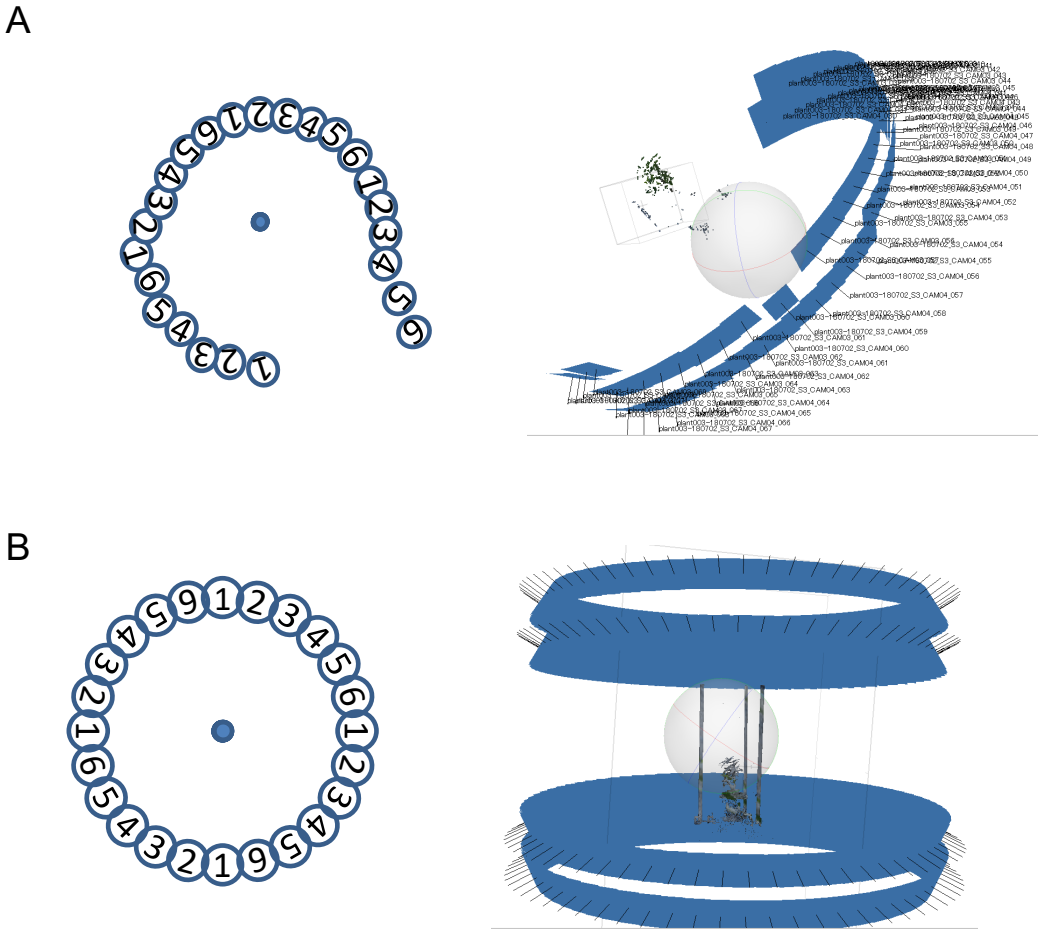


Figure 10. A risk of camera position estimation failure due to the accumulation of camera position estimation errors for all-around images taken at short intervals in SfM

1

GmJMC092 (KUROHIRA)

HOJAK

GmJMC112 (FUKUYUTAKA)

MISUZUDAIZU

Supplemental Figure S1
Results of reconstruction of 3D point clouds of four soybean varieties using the proposed method.

The PLY data in this figure can be downloaded via "https://github.com/tanasoft/MMM-CLCFM"

GmJMC092 (KUROHIRA)

HOJAK

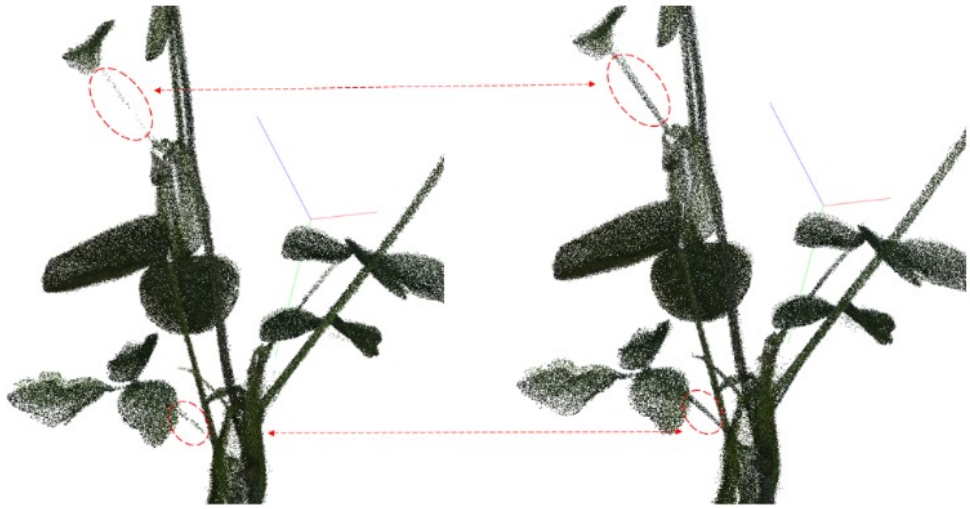
GmJMC112 (FUKUYUTAKA)

MISUZUDAIZU

Supplemental Figure S2.
Results of reconstruction of 3D point clouds for the same image in Fig. S1 using Metashape.

1

A

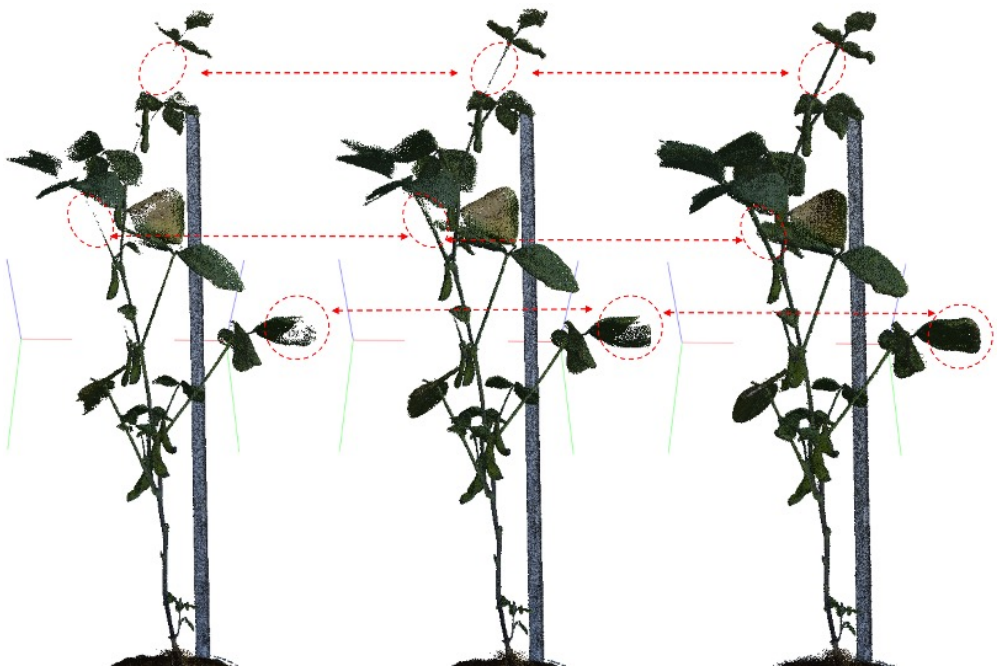


Executed MVS used four images
(Viewing angle 15 degree)

Executed MVS used two images
(Viewing angle 5 degree)

Difference in the number of images used in MMM

B



Viewing angle
240 degree

Viewing angle
180 degree

Viewing angle
90 degree

Difference in the number of mask images used in getting rid of points process

Supplemental Figure S3.
Results of running the analysis in Fig. 5, Fig. 6 using Metashape 1.6

1

MMM process without the point removal process

MMM process with the point removal process

Supplemental Figure S4.
Results of running the analysis in Fig. 7 using Metashape 1.6

1