Descriptor-Free Multi-View Region Matching for Instance-Wise 3D Reconstruction

Takuma Doi[1], Fumio Okura[1,2][0000−0001−7595−1300], Toshiki Nagahara[1],
Yasuyuki Matsushita[1][0000−0002−1935−4752], and Yasushi Yagi[1]

[1] Graduate School of Information Science and Technology,
Osaka University, Osaka, Japan
{doi,t-nagahara,yagi}@am.sanken.osaka-u.ac.jp
{okura,yasumat}@ist.osaka-u.ac.jp
[2] Japan Science and Technology Agency, Saitama, Japan

Abstract. This paper proposes a multi-view extension of instance segmentation without relying on texture or shape descriptor matching. Multi-view instance segmentation becomes challenging for scenes with repetitive textures and shapes, *e.g.*, plant leaves, due to the difficulty of multi-view matching using texture or shape descriptors. To this end, we propose a multi-view region matching method based on epipolar geometry, which does not rely on any feature descriptors. We further show that the epipolar region matching can be easily integrated into instance segmentation and effective for instance-wise 3D reconstruction. Experiments demonstrate the improved accuracy of multi-view instance matching and the 3D reconstruction compared to the baseline methods.

1 Introduction

One of the important applications of computer vision is to analyze the shape and structure of natural objects, *i.e.*, quantifying the size, number, and shape of plant leaves. In such applications, *instance segmentation* plays an important role because of the need for accurately determining individual object instances. A single-view instance segmentation has gained attention, and there have been a series of previous works [1–3] to address the problem. However, for scenes suffering from severe occlusions, it is needed to perform instance segmentation in a *multi-view* setting as shown in Fig. 1. We call it a *multi-view instance segmentation (MVIS)* problem.

An extension of single-view instance segmentation to the multi-view setting is a challenging task, especially when the individual instances are look-alike, for example, plant leaves. Although external markers or background objects, *e.g.*, plant pots, can be a guide for obtaining sparse correspondences to determine camera poses, we cannot expect dense point correspondences on similar instances, *e.g.*, leaves, across views. This setting makes it difficult to directly use existing approaches, such as 3D instance segmentation [4] and multi-view semantic segmentation [5–7], which heavily rely on 3D shape information.

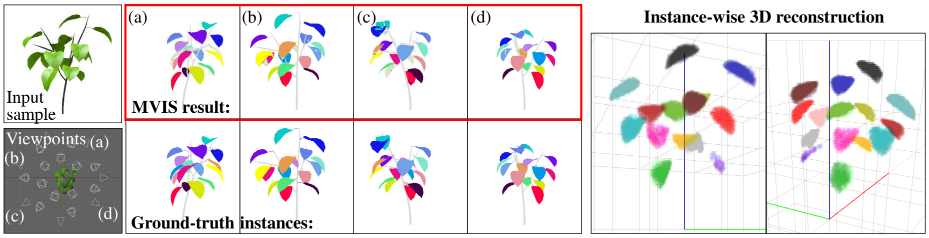

Fig. 1: An example of MVIS and instance-wise 3D reconstruction. The same color indicates the corresponding instances across different views. We propose a multi-view instance matching without relying on texture or shape descriptors.

To overcome the problem, we propose an MVIS method based on region matching that does not rely on dense correspondences, which can be used for matching objects across views without distinctive textures or shape but with similar appearances. The key idea of our method is to use the epipolar constraint to determine the multi-view region correspondences of segmented instances that are computed at each view. By treating each instance in each view as a node in a graph, and by assigning edge weights based on the degree of intersection of epipolar lines and instance regions, we cast the problem of multi-view instance segmentation to a graph clustering problem. We call the approach the *epipolar region matching*. We show that it enables reliably establishing correspondences of view-wise segmented instances across multiple views. Further, our epipolar region matching can be easily integrated in modern instance segmentation methods [1, 3, 8]. Our method can be combined with any region segmentation methods; while given a set of regions, our region matching method does not rely on any descriptors. The estimated instances with multi-view correspondences can be used for the *instance-wise 3D reconstruction* via existing 3D reconstruction methods, yielding superior reconstruction results.

Experimental results show that the proposed method yields accurate multi-view instance correspondence compared to an instance matching method based on a traditional multi-view stereo (MVS) and a point-based matching. We also demonstrate the effectiveness of MVIS for instance-wise 3D reconstruction.

Contributions. We propose a multi-view region matching method without using texture and shape descriptors, as a key technical component of MVIS for texture/shape-repetitive scenes. The matching method can be put together in instance segmentation and 3D reconstruction methods for instance-wise 3D reconstruction.

2 Related Works

Our work is closely related to multi-view correspondence matching, as well as the multi-view or 3D extensions of semantic/instance segmentation. This section discusses the related works in these subject areas.

Multi-view correspondence matching. Multi-view correspondence matching is a fundamental problem in multi-view image analysis, such as structure from motion (SfM) [9] and multi-view stereo (MVS) [10]. As a veteran but still frequently used approach, keypoint detection and matching (*e.g.*, [11–13]) is used to obtain sparse correspondences; while dense correspondence matching often involves a patch [14] and plane-based [15] matching.

For target scenes containing partly texture-less objects, various approaches have been studied, such as a belief propagation [16, 17] for MVS. More recently, deep-learning-based approaches to correspondence matching for partially texture-less scenes, which often assume the smoothness of the target scene, have been studied [18, 19]. Correspondence matching for objects in which surface is fully texture-less, or with repetitive textures is notably challenging because it becomes difficult to find dense correspondences. For string-like objects (*e.g.*, hairs), which have the matching ambiguity along the strings or lines, an MVS method using line-shaped patches has been proposed [20]. Regarding the sparse point matching, Dellaert *et al.* [21] presented a method for the SfM problem without using texture features, which calculates the camera pose and sparse point correspondences based only on the geometric relationship of keypoints.

Shape-based matching [22–24] such as using the Fourier shape descriptors [25] or shape contexts [22], is another line of the region matching. These approaches encode the region shape information, *e.g.*, instance shape, into descriptors. In our cases, since the target scene is composed of objects with similar 3D shapes (*e.g.*, plant leaves), it is not realistic to use shape-based matching. Although it is different from the context of instance segmentation, region-based information [26], such as the statistics of textures, is a useful cue for the correspondence matching. For example, superpixel stereo [27, 28] or segment-based stereo [29] were developed to increase the robustness of correspondence matching. Segmentation of the same object among multiple views can be categorized in multi-view co-segmentation [5–7], which also uses the texture features.

Unlike the previous methods, we achieve the descriptor-free multi-view region matching using the epipolar constraint. The epipolar matching is a traditional yet important problem. As far as we are aware, *searching regions without (texture or shape) descriptor matching* remains unsolved.

Instance segmentation and its multi-view extension. While various implementations of instance segmentation including one-stage methods [30] are developed, two-stage frameworks that combine object detection and mask generation are often used due to the advantage of performance and simplicity. A major implementation of two-stage instance segmentation is Mask R-CNN [1] based on Faster R-CNN [31], which computes the region proposals of the target objects followed by the mask generation for selected proposals.

Video instance segmentation is a multi-image extension of instance segmentation. These approaches perform the instance tracking, as well as the segmentation [32–35]. An early attempt of this approach [36] used superpixels for instance tracking. A recent study configures the problem as multi-object tracking and segmentation (MOTS), and provides a Mask-R-CNN-based implementation [8].

PanopticFusion [37] aggregates multi-view instances in a 3D volumetric space, while it inputs a video sequence that enables the instance tracking. Video instance tracking is applied for plant image analysis, such as the 3D reconstruction of grape clusters [38]. This batch of approaches assumes the time-series image sequences, which allows the object tracking with smaller displacement between the consequent frames. We rather focus on the multi-view setting, where the images are captured with wider baselines between the viewpoints.

3D semantic/instance segmentation. Somewhat related to MVIS, 3D semantic/instance segmentation is studied. Approaches for 3D segmentation often input 3D shapes of target objects, such as point cloud [39, 40], volumetric space [41], and RGB-D images [4, 42]. Semantic segmentation on multi-view RGB images can be used for 3D reconstruction via the aggregation on a 3D space, which are called semantic 3D reconstruction [43–45]. These methods do not distinguish the objects in a same category in principle.

Nassar *et al.* [46] proposes an instance warping for multi-view instance correspondence matching using the 3D shape of target scenes. Instance segmentation on 3D point clouds are also studied [47–49], which needs the input of high-quality 3D point clouds. Unlike these approaches, we focus on the instance matching of objects where the 3D reconstruction is challenging; *e.g.*, plants due to thin shape, repetitive textures, and heavy occlusions. As an application-specific study, a plant modeling method proposed by Quan *et al.* [50] uses the combination of 2D segmentation and 3D point cloud clustering for 3D leaf modeling. Because their method assumes a good-quality 3D point cloud as input, it is difficult for the reconstruction of plants with texture-less leaves.

3 Multi-view instance segmentation (MVIS)

Our goal is to find multi-view instance correspondences from the multi-view images of the target. We assume that the camera poses and intrinsics are known, *e.g.*, by an SfM [9] using the sparse correspondences obtained from the scene background. We here assume that instance segmentation in each view is available for now, while the instance correspondence *across views* is unknown.

3.1 Epipolar region matching

Given a set of instances in each image and camera pose information, we establish instance correspondences across multi-view images. We approach this problem by epipolar geometry; namely, finding the correspondences via epipolar lines drawn on other views. Since multiple instances appear on the same epipolar line (see Fig. 2), we cast the matching problem to a *graph clustering* problem.

We create an undirected edge-weighted graph $G = (V, E, w)$, where V is the node set that consists of instances appeared in all views, $E \subseteq V \times V$ is the edge set, and $w : E \to \mathbb{R}_+$ is the weight function defining the edge weights. The number of nodes $|V|$ corresponds to the total number of instances that appear in all the images.

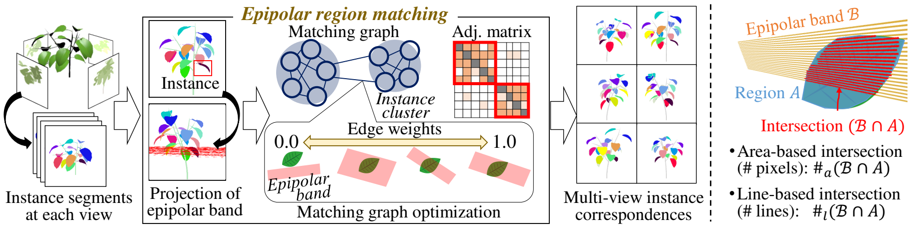
Epipolar region matching
Matching graph
Adj. matrix
Instance
Instance cluster
Edge weights
0.0
1.0
Epipolar band
Instance segments at each view
Projection of epipolar band
Matching graph optimization
Multi-view instance correspondences
Epipolar band $\mathcal{B}$
Region A
Intersection ($\mathcal{B} \cap A$)
• Area-based intersection (# pixels): $\#_a(\mathcal{B} \cap A)$
• Line-based intersection (# lines): $\#_l(\mathcal{B} \cap A)$

Fig. 2: Overview of epipolar region matching which does not rely on the feature descriptors. The proposed method creates a matching graph based on the epipolar constraint, where the edge weights are defined as the degree of intersection between the instances and epipolar bands (*i.e.*, projections of instances in other views). Instance correspondences are extracted by a graph clustering method.

To define the edge weights, we compute a set of epipolar lines from densely-sampled points on each instance. Here we define the i-th view image as I_i, m-th instance segment in the i-th image as $v_{i,m} \in V$ and its region (set of pixels) in the image coordinates as $\mathcal{A}_{i,m} \subset \mathbb{R}^2$. Let $\mathbf{F}_{ij}$ be a fundamental matrix between i-th and j-th views. Then an image point $\mathbf{p}$ in instance segment $\mathcal{A}_{i,m}$, *i.e.*, $\mathbf{p} \in \mathcal{A}_{i,m}$, forms an epipolar line $\mathbf{l}_j(\mathbf{p})$ in the j-th view image as

$$\mathbf{l}_j(\mathbf{p}) = \mathbf{F}_{ij}\tilde{\mathbf{p}}, \tag{1}$$

where $\tilde{\mathbf{p}}$ is the homogeneous representation of $\mathbf{p}$. From all points in region $\mathcal{A}_{i,m}$, we have a set of epipolar lines forming a pencil of epipolar lines passing through the epipole. In what follows, we call the pencil an *epipolar band* $\mathcal{B}$ in this paper. The epipolar band $\mathcal{B}$ of region $\mathcal{A}_{i,m}$ on the j-th image is thus defined as

$$\mathcal{B}(\mathcal{A}_{i,m}, I_j) = \{\mathbf{F}_{ij}\tilde{\mathbf{p}}\}, \quad \mathbf{p} \in \mathcal{A}_{i,m}. \tag{2}$$

We define the edge weights w in the graph G as the degree of intersection of an epipolar band $\mathcal{B}$ and instance regions $\mathcal{A}$ in analogous to the intersection-over-union (IoU) computation. While original IoU is defined between two areas, the extension to a similarity measure between the epipolar band and a region is not straightforward. To evaluate the degree of intersection between the epipolar band and instance region, we use two measures: The area of intersection and the number of epipolar lines in the epipolar band passing through the instance region, as illustrated in the right side of Fig. 2. In this manner, the edge weight w between nodes $v_{i,m}$ and $v_{j,n}$ can be obtained as

$$w(v_{i,m}, v_{j,n}) = \frac{\#_a(\mathcal{B}(\mathcal{A}_{i,m}, I_j) \cap \mathcal{A}_{j,n})}{\#_a\mathcal{A}_{j,n}} \cdot \frac{\#_l(\mathcal{B}(\mathcal{A}_{i,m}, I_j) \cap \mathcal{A}_{j,n})}{\#_l\mathcal{B}(\mathcal{A}_{i,m}, I_j)}, \tag{3}$$

where the function $\#_a$ counts the number of pixels belonging to the region, while $\#_l$ counts the number of epipolar lines passing through the area. During the computation of $\#_a(\mathcal{B}(\mathcal{A}_{i,m}, I_j) \cap \mathcal{A}_{j,n})$, we draw epipolar lines with the same thickness (two pixels was used in our experiment). As with many other similarity measures, $w(v_{i,m}, v_{j,n})$ takes the range $[0, 1]$, where it becomes one if all epipolar lines in the epipolar band passes through the instance area and a whole part of the area is filled with epipolar lines (see Fig. 2).

Once the edge weights $\{w\}$ are defined, we form an adjacency matrix $\mathbf{W}$ of $|V| \times |V|$ elements from the graph G and perform a graph clustering using symmetric non-negative matrix factorization (SymNMF) [51], as $\mathbf{W} \approx \mathbf{H}\mathbf{H}^T$. After the factorization, the largest element in each row of $\mathbf{H}$ indicates the *cluster ID*, where each cluster forms a set of corresponding instances across multi-view. Hereafter, we call the cluster an *instance cluster*.

3.2 Application: Epipolar region matching for region proposals

As an application of the proposed method, we here describe the integration of multi-view region matching method with modern instance segmentation. While our region matching method can be used with any instance segmentation methods, we can optionally utilize the correspondences across multi-view *region proposals* to retain partly occluded instances. Most instance segmentation methods [1, 3, 8] rely on region proposals [31], which performs region unification by the non-maximum suppression (NMS) [52] to merge overlapping region proposals. As a result, region proposals for partially occluded instances are often suppressed. To this end, we implemented an NMS process considering the multi-view region correspondences. We found our implementation recovered partially occluded instances, while it did not significantly affect the overall instance segmentation accuracy (see the supplementary material for detailed discussions).

Our method starts from the initial segmentation result, *e.g.*, by Mask R-CNN, and their tentative multi-view correspondence computed by the epipolar region matching. The NMS and region matching processes can be performed alternately to update the set of detected instances and their multi-view correspondences as illustrated in Fig. 3. We assign the instance cluster IDs for each region proposal by the region matching method, and avoid to unify the proposals with different cluster IDs by the NMS process.

To update the cluster ID of each region proposals, sets of corresponding instances (*i.e.*, instance clusters estimated in the previous iteration) are projected to the view using the same manner as the epipolar band projection described in Sec. 3.1. Let the k-th instance cluster in the previous iteration be $\{\mathcal{A}_k\} \subset \{\mathcal{A}\}$, which is denoted as a set of instances. The projection onto the i-th image, $\mathcal{M}_{i,k}$, is calculated based on the sum of epipolar bands by the instances (we call $\mathcal{M}_{i,k}$ an *epipolar map* hereafter).

$$\mathcal{M}_{i,k} = \frac{\sum_{\{\mathcal{A}_k\}} \mathcal{B}(\mathcal{A} \in \{\mathcal{A}_k\}, I_i)}{\#_c\{\mathcal{A}_k\}}, \tag{4}$$

where $\#_c\{\mathcal{A}_k\}$ counts the number of instances in the cluster $\{\mathcal{A}_k\}$. The right side of Fig. 3 shows an example of the epipolar map.

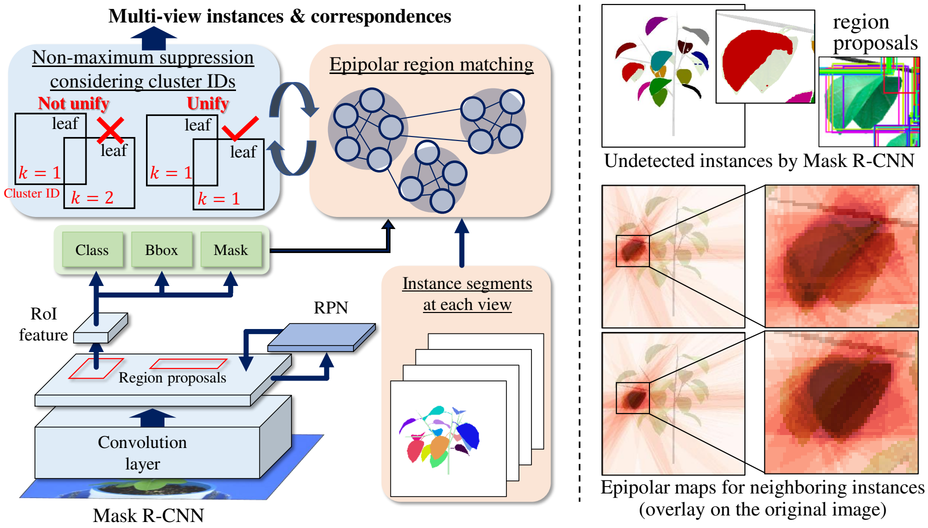
Multi-view instances & correspondences
Non-maximum suppression considering cluster IDs
Not unify
Unify
leaf
leaf
$k = 1$
Cluster ID
$k = 2$
leaf
leaf
$k = 1$
$k = 1$
Epipolar region matching
Class
Bbox
Mask
RoI feature
RPN
Region proposals
Convolution layer
Mask R-CNN
Instance segments at each view
region proposals
Undetected instances by Mask R-CNN
Epipolar maps for neighboring instances (overlay on the original image)

Fig. 3: Epipolar region matching integrated in a two-stage instance segmentation network. It updates the instance detection and multi-view correspondence via an iterative process. The projections of epipolar bands are used to distinguish the undetected region proposal from neighboring instances.

An updated cluster ID for each region proposal is calculated based on the highest degree of intersection between the epipolar map and the region proposal. Letting the instance mask of r-th region proposal in i-th image as $\mathcal{A}'_{i,r}$, the similarity of the two maps, $\mathcal{M}_{i,m}$ and $\mathcal{A}'_{i,r}$, is obtained by the similar manner to the IoU computation. Since the epipolar map does not take the value of $\{0, 1\}$, we use an extension of the IoU, which is called Ruzicka similarity [53].

$$s(\mathcal{M}_{i,k}, \mathcal{A}'_{i,r}) = \frac{\Sigma_{\mathbf{x}} \min(\mathcal{M}_{i,k}(\mathbf{x}), \mathcal{A}'_{i,r}(\mathbf{x}))}{\Sigma_{\mathbf{x}} \max(\mathcal{M}_{i,k}(\mathbf{x}), \mathcal{A}'_{i,r}(\mathbf{x}))}, \tag{5}$$

where $\mathbf{x}$ denotes the pixel location. In the equation, we deal with each instance mask $\mathcal{A}'_{i,r}$ as a map taking $\{0, 1\}$ values, where the value takes one if the pixel is inside the mask. The updated cluster ID for the region proposal $\mathcal{A}'_{i,r}$ is selected as k with the largest similarity $s(\mathcal{M}_{i,k}, \mathcal{A}'_{i,r})$. The update process of cluster ID is followed by the NMS that does not unify the region proposals with different cluster IDs to retain the partly occluded instances. Our implementation allows to iterate the region matching and NMS processes while updating the set of instances and multi-view correspondences.

3.3 Application: Instance-wise 3D reconstruction

Multi-view correspondences of instance segments can be used for the *instance-wise* 3D reconstruction, by independently applying a 3D reconstruction method for each of instance clusters. We implemented a simple volumetric reconstruction method based on a back-projection used in a traditional computed tomography [54], which is analogous to the visual hull method [55].

Let a set of k-th instance cluster $\{\mathcal{A}_k\}$ and the set of projection functions $\{\theta\}$ from 3D to 2D image coordinates, $\theta : \mathbb{R}^3 \to \mathbb{R}^2$. The aggregated value $\mathcal{I}_k$ at voxel $\mathbf{x}_{3D} \in \mathbb{R}^3$ can be computed as:

$$\mathcal{I}_k(\mathbf{x}_{3D}) = \frac{\sum_{\mathcal{A}_k \in \{\mathcal{A}_k\}} \mathcal{L}_{\mathcal{A}_k}(\theta_k(\mathbf{x}_{3D}))}{\#_c\{\mathcal{A}_k\}}, \tag{6}$$

in which θ_k represents a projection from the voxel to the image coordinates corresponding to the instance $\mathcal{A}_k$. Here, $\mathcal{L}_{\mathcal{A}_k}$ denotes a mask representation of $\mathcal{A}_k$, which returns 1 if the pixel is inside the instance region $\mathcal{A}_k$. The resultant voxel space $\mathcal{I}_k$ represents the ratio of voted instances; while we simply yielded the binarized version of the voxel space for the evaluation of the reconstruction accuracy, using the threshold of $\mathcal{I}_k(\mathbf{x}_{3D}) = 0.5$.

3.4 Implementation details

Region matching. During the epipolar band projection, we randomly sampled 200 points in each instance to draw the epipolar lines. The graph clustering is based on a Python implementation of SymNMF [51]. Since the clustering algorithm requires the number of instance clusters $|k|$ as an input, we implemented a framework to search the optimal number of clusters. Assuming the instances are evenly occluded and are projected onto the similar number of views, we selected the optimal $|k|_{opt}$ with the minimum standard deviation of the number of instances contained in the clusters.

$$|k|_{opt} = \underset{|k|}{\operatorname{argmin}} \sqrt{\sum_{\mathcal{A}_k} (\#_c\{\mathcal{A}_k\}_{mean} - \#_c\{\mathcal{A}_k\})^2}, \tag{7}$$

where $\#_c\{\mathcal{A}_k\}_{mean}$ denotes the mean number of instances in instance clusters. In the supplementary material, we provide a detailed analysis when the number of instance clusters $|k|$ is given (*i.e.*, using the ground-truth number of objects).

MVIS application. For the integration of the region matching for region proposals, we used a Keras implementation[3] of Mask R-CNN. We implemented an NMS using `numpy` and `nms` package outside the computation graph of Mask R-CNN. To obtain the initial instances, we used the original Mask R-CNN with NMS by a large RoI threshold (0.7 in our experiment) and obtained excessive numbers of object RoIs with their instance masks. During the iterations, our NMS is performed for each instance ID independently with a smaller threshold (0.3 was used in the experiment). With our unoptimized implementation, the whole process took up to several hours on a CPU (2.1 GHz, 8 threads); the projection of epipolar bands spent most of the time, which should be greatly optimized through a better implementation.

[3] `https://github.com/matterport/Mask_RCNN`

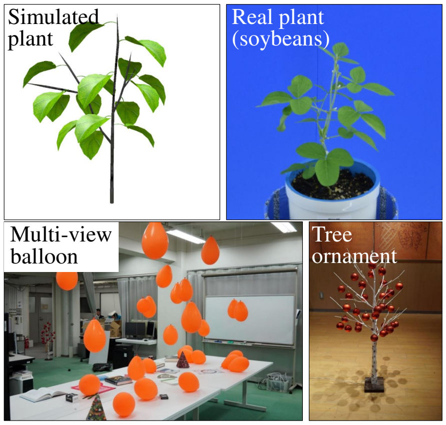

Fig. 4: Datasets used in the experiment.

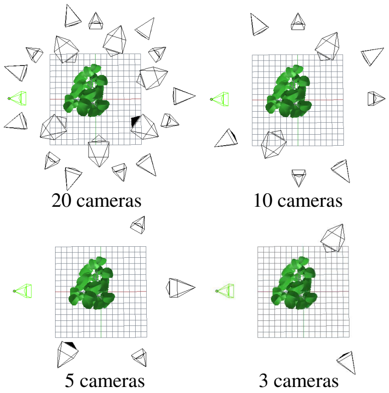

Fig. 5: Camera setting for simulated plant dataset.

4 Experiments

We conducted experiments to assess the quality of the multi-view matching and instance-wise 3D reconstruction, which are core part of our framework. The supplementary material provides the detailed analysis and discussions, including the effect on the instance detection by our implementation.

4.1 Dataset

We used the following datasets for the experiment (samples are shown in Fig. 4).

Simulated plant dataset. We used simulated plant models, which were modified from a dataset used in a plant modeling study [56]. This dataset contains the ground truth instance masks and their multi-view correspondences, as well as the 3D shape information. For training of Mask R-CNN, we prepared 14 simulated plant models rendered from $32,760$ viewpoints ($458,640$ images in total). For evaluation, we used four plant models with a different number of leaves (4, 8, 16, and 32), not used for the training. To assess the effect of the varying number of views, we rendered the plants from different number of viewpoints (3, 5, 10, and 20) illustrated in Fig. 5, where we used the ground truth camera parameters.

Real-world dataset. We prepared the real-world scenes for experiment, in which the ground truth instance masks and multi-view correspondences were manually created. The camera poses for these scenes were estimated via SfM [9].

Real plant (soybeans) scene is a set of multi-view images of a soybean plant captured by a multi-view capturing system [57], which was originally created for plant science studies. We used 28 views for instance matching. The number of distinctive leaves in the plant was 28. COCO-trained Mask R-CNN was fine-tuned using the images of eight soybeans plants captured by the same system, which were not used during the evaluation.

Multi-view balloon scene contains a scene with 27 balloons captured from 18 views. Mask R-CNN, pre-trained with the COCO dataset, was fine-tuned using 61 balloon images in the balloon dataset[4].

Tree ornament scene contains an artificial tree with 33 decorations captured from 21 views. Because the ornaments have spherical shapes, we fine-tuned the balloon-trained Mask R-CNN using a small number of (*i.e.*, 25) training images with tree ornaments, which were not used in the evaluation.

4.2 Multi-view matching results

We evaluated the accuracy of multi-view region matching using the dataset.

Baselines. Because of no established baseline methods for descriptor-free region matching, we compared the proposed approach with straightforward implementations of instance matching 1) using MVS-based point cloud as guidance, 2) using the center point of regions, and 3) using sparse correspondences.

MVS-based matching. 3D model by traditional MVS should be a guide for multi-view matching. We implemented a baseline matching method using dense 3D point clouds, called *MVS-based matching* hereafter. In this implementation, we used dense 3D point clouds reconstructed by COLMAP [10]. For $\mathcal{A}_{i,m}$, the m-th instance in the i-th view, we selected a 3D position $\mathbf{p}_{\mathcal{A}_{i,m}}$ that represents the region of $\mathcal{A}_{i,m}$. An straightforward way to select $\mathbf{p}_{\mathcal{A}_{i,m}}$ that is closest to the camera from among the points projected in the region of $\mathcal{A}_{i,m}$. To increase the robustness to the outliers, we designated $\mathbf{p}_{\mathcal{A}_{i,m}}$ as the centroid of the set of top 10% closest points among those projected in the region.

We then project $\mathbf{p}_{\mathcal{A}_{i,m}}$ to the other views; for example, if $\mathbf{p}_{\mathcal{A}_{i,m}}$ is projected on the n-th instance in the j-th view, $\mathcal{A}_{j,n}$, we deemed $\mathcal{A}_{i,m}$ and $\mathcal{A}_{j,n}$ are corresponding instances. In the same manner to our proposed method, we constructed a matching graph with $|V|$ nodes, where the edge weights record the correspondences; *e.g.*, the edge between $\mathcal{A}_{i,m}$ and $\mathcal{A}_{j,n}$ are weighted by one. We solved the graph clustering using SymNMF [51]. This baseline method expects that a good-quality 3D model is given, which is under the same assumption as the methods using point clouds as an input (*e.g.*, [50]).

Point-based matching. Another baseline of the proposed approach is to match the centroid of instance regions instead of using the region matching. This method is analogous to the traditional *point-based* epipolar matching, *i.e.*, methods using epipolar *lines* instead of *bands*. In our implementation, an epipolar line is drawn for the centroid of each instance region. The correspondences between instances in different views can be yielded as the nearest epipolar lines in 3D space. Similar to the MVS-based method, we constructed a matching graph, where the edge weights between the corresponding instances are weighted by one, and solved the graph clustering by SymNMF.

[4] `https://github.com/matterport/Mask_RCNN/tree/master/samples/balloon`

Matching based on sparse feature correspondences. We also assessed a multi-view region matching method using sparse correspondences, which is a typical example of feature descriptors. We used AKAZE keypoints and descriptors [12]. For each instance region, a corresponding instance for each different view is searched based on the number of corresponding points within the regions. We solved the graph matching similarly with the other baseline methods. This method is expected to work well when the scene is observed from similar viewing angles, which is a similar assumption to the video instance tracking.

Evaluation metric. The matching accuracy s_{match} was calculated by the number of correctly classified instances over the number of all instances $|V|$. Because we solve the multi-view matching as a clustering problem, we need to associate estimated instance clusters and the ground-truth clusters. We determined the ground-truth cluster ID k_{gt} corresponding to the ID of an estimated cluster k_{est}, by searching the mode of k_{gt} among instances belonging to k_{est}.

$$s_{match} = \frac{\sum_{k_{est}} \max_{k_{gt} \in \{k_{gt}\}} |\mathrm{id}(\mathcal{A}_{k_{est}}) \cap k_{gt}|}{|V|}, \tag{8}$$

where $\mathrm{id}(\mathcal{A}_{k_{est}})$ denotes the estimated ID of the instance $\mathcal{A}_{k_{est}}$, while $\{k_{gt}\}$ is the set of ground-truth instance cluster ID. Therefore, $|\mathrm{id}(\mathcal{A}_{k_{est}}) \cap k_{gt}|$ returns 1 if the two instance IDs are same. The number of *correct* matches is divided by $|V|$, the total number of instances among multi-view images. As an evaluation of clustering problems, s_{match} is equivalent to the *purity* metric, which is a common measure for evaluating the success of clustering.

We also evaluated the accuracy of the estimated number of clusters $|k|_{opt}$ by computing the mean absolute error (MAE) $e_{|k|}$ of the estimated number of clusters via Eq. (7).

Result. Table 1 shows the the matching accuracy s_{match} and the MAE of cluster numbers $e_{|k|}$ for the simulated plant dataset. The table compares the accuracy and error averaged over different number of cameras and leaves. The proposed MVIS implementation yielded a better accuracy for most cases and achieved the average matching accuracy of 88.3%, which outperforms the accuracy by the baseline methods. Also, the result shows MVIS accurately estimates the number of clusters (*i.e.*, the number of objects in the scene). MVS-based matching yielded better accuracy when using a larger number of cameras (*i.e.*, 20) because the smaller differences in the viewing angles enable it to find the dense texture-based correspondences. In the cases using a smaller number of views, the MVS-based method notably drops the performance, although the proposed approach still achieved the matching accuracy of over 85% when the number of views is 3.

Table 2 shows the matching performance for the real-world scenes. The proposed approach also achieved a reasonable accuracy for both multi-view matching and cluster number estimation. For the tree ornament scenes, the matching accuracy was comparable among the methods. Ornaments were small spheres, and well approximated as a point. Proposed MVIS has an advantage for the scenes with difficulties of dense 3D reconstruction (*e.g.*, real plant scene) or with relatively large objects (*e.g.*, balloon scene).

Table 1: Multi-view matching accuracy s_{match} and MAE of the number of clusters $e_{|k|}$ for CG plant dataset. The values are averaged over the number of cameras and leaves. (Bold: the best accuracy among the methods.)

	Method	# cameras				# leaves				Average
		20	10	5	3	32	16	8	4	
s_{match} ↑	MVS-based	**0.935**	0.735	0.211	0.215	0.337	0.584	0.543	0.631	0.524
	Point-based	0.715	0.697	0.521	0.463	0.477	0.497	0.612	0.810	0.599
	Sparse	0.392	0.488	0.562	0.509	0.376	0.455	0.500	0.620	0.488
	MVIS	0.905	**0.910**	**0.860**	**0.856**	**0.655**	**0.895**	**0.987**	**0.994**	**0.883**
$e_{\|k\|}$ ↓	MVS-based	**1.50**	7.25	13.25	9.25	17.00	10.00	3.25	1.00	7.81
	Point-based	19.50	25.75	4.00	8.50	11.75	16.75	20.25	9.00	14.44
	Sparse	31.00	26.00	21.50	9.25	15.25	25.75	22.25	24.50	21.94
	MVIS	2.00	**2.25**	**3.00**	**2.50**	**9.00**	**0.75**	**0.00**	**0.00**	**2.44**

Table 2: Multi-view matching accuracy s_{match} and MAE of number of clusters $e_{|k|}$ for real-world dataset. (Bold: the best accuracy among the methods.)

Method	Real plant		Balloon		Ornament		Average	
	s_{match} ↑	$e_{\|k\|}$ ↓	s_{match} ↑	$e_{\|k\|}$ ↓	s_{match} ↑	$e_{\|k\|}$ ↓	s_{match} ↑	$e_{\|k\|}$ ↓
MVS-based	0.220	36	0.081	25	0.680	**1**	0.327	20.7
Point-based	0.239	34	0.545	40	**0.683**	34	0.489	36.0
Sparse	0.237	41	0.270	36	0.679	33	0.395	36.7
MVIS	**0.562**	**26**	**0.751**	**3**	0.676	**1**	**0.663**	**10.0**

4.3 Instance-wise 3D reconstruction results

We here describe the quantitative result of 3D reconstruction using simulated plant models, which we have the ground-truth shape of the leaves.

Baselines. We used two baselines for the evaluation of 3D reconstruction accuracy, although these methods do not provide the instance-wise 3D reconstruction. Since we used back-projection-based 3D reconstruction described in Sec. 3.3, we implemented a simple method using the back projection. Without relying on the instance correspondence, we unified the silhouettes of all leaves and inputted to the back projection. This mimics the *semantic segmentation* of leaves as the silhouette source, instead of using multi-view instances. As another baseline, we simply used the *MVS-based point clouds* reconstructed by COLMAP [10].

Evaluation metric. We evaluated the geometric error between the dense point clouds of the reconstructed and the ground truth leaf shapes. For the evaluation, we first unify all 3D shapes of reconstructed leaf instances and convert the 3D voxel representation to a dense point cloud, where a 3D point is located if the voxel is inside leaves. The ground-truth leaf shape was originally modeled using polygons, we oversampled the vertices by Catmull–Clark subdivision [58] to yield the dense point clouds.

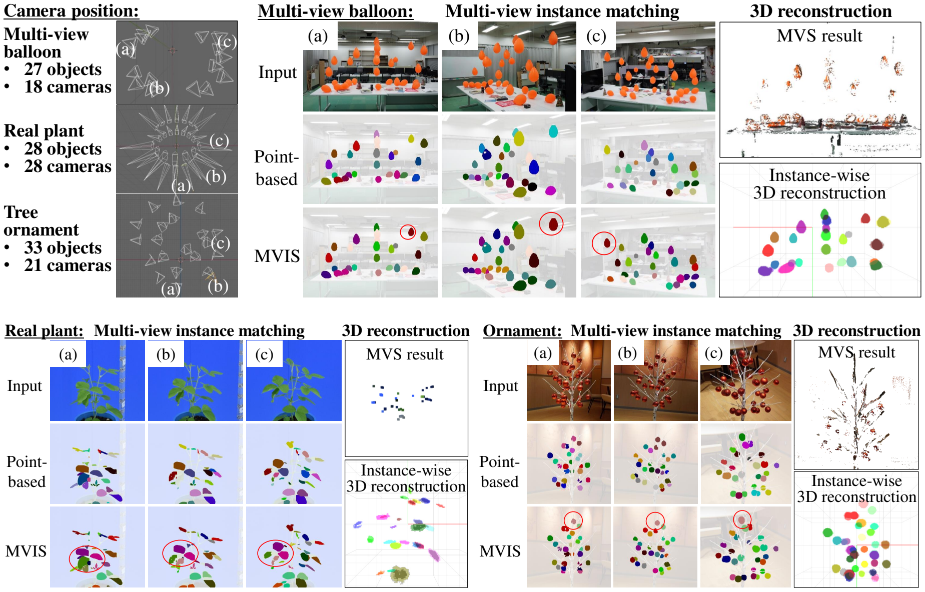

Fig. 6: Visual results for real-world examples. We show multi-view instances via the proposed MVIS and the point-based matching, which yielded a better matching accuracy than the other baselines. Although some instances are undetected or misclassified, MVIS yields convincing correspondences (*e.g.*, the circled objects) and 3D reconstructions.

Let $\mathbf{i} \in \mathcal{I}$ and $\mathbf{t} \in \mathcal{T}$ be estimated and the ground-truth 3D points, respectively. The geometric error is defined as a bidirectional Euclidean distance [59] between the two point sets written as

$$d(\mathcal{I}, \mathcal{T}) = \frac{1}{2}\left(\frac{\sum_{\mathcal{I}} ||\mathbf{i} - N_{\mathcal{T}}(\mathbf{i})||}{|\mathcal{I}|} + \frac{\sum_{\mathcal{T}} ||\mathbf{t} - N_{\mathcal{I}}(\mathbf{t})||}{|\mathcal{T}|}\right),$$

where $N_{\mathcal{I}}(\mathbf{x})$ and $N_{\mathcal{T}}(\mathbf{x})$ are functions to acquire the nearest neighbor point to $\mathbf{x}$ from point sets $\mathcal{I}$ and $\mathcal{T}$, respectively, and $|\mathcal{I}|$ and $|\mathcal{T}|$ denote the numbers of points in $\mathcal{I}$ and $\mathcal{T}$.

Results. Table 3 shows the 3D reconstruction error $d(\mathcal{I}, \mathcal{T})$. Since the geometric error is defined only up to scale like most multi-view 3D reconstruction methods, the errors were normalized by the average leaf length. For the comparison across the different number of cameras (left side of the table), we averaged the error over the different number of leaves. For the right half of the table, results using 10 cameras are listed because MVS often failed the dense reconstruction for the smaller number of views. The proposed method (MVIS) achieved better accuracy in most cases. Although the traditional MVS yielded an accurate reconstruction when using a larger number of (*i.e.*, 20) cameras or smaller number of leaves,

Table 3: 3D reconstruction error $d(\mathcal{I}, \mathcal{T})$ normalized by the mean leaf length. N/A denotes the failure of the reconstruction (*i.e.*, no point cloud was generated).

Method	# cameras (averaged)				# leaves (10 cameras)			
	20	10	5	3	32	16	8	4
MVS (COLMAP)	**0.027**	0.125	N/A	N/A	0.311	0.111	**0.046**	**0.030**
Semantic segmentation	1.593	1.707	2.556	2.068	1.306	2.269	1.396	1.858
MVIS (proposed)	0.077	**0.082**	**0.185**	**0.129**	**0.183**	**0.057**	**0.046**	0.042

the reconstruction was inaccurate or failed due to the difficulties of finding dense correspondences. The average reconstruction error by the proposed MVIS did not notably drop when decreasing the number of views, which still achieved the error of 12.9 % of the average leaf length via the reconstruction using 3 cameras.

Visual examples. Figure 6 shows example results of MVIS and instance-wise 3D reconstruction for real-world datasets. In the MVIS result, corresponding instances are visualized with the same color. The proposed method yields the multi-view correspondences and instance-wise 3D shapes convincingly, although we do not have access to the ground-truth 3D shapes for real-world datasets.

Failure cases. Because the proposed method relies on the region segmentation, the failure in segmentation due to *e.g.*, contact/occlusions of objects, affects the matching accuracy. Our experiment includes such cases when increasing the number of objects (*e.g.*, Fig. 7). The low quality of instance masks is the dominant cause of the mismatching, which is a limitation of the proposed method.

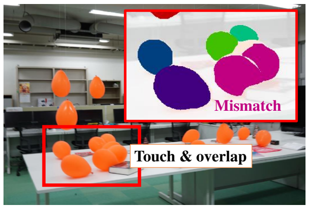

Fig. 7: Failure case.

5 Discussion

We introduced a multi-view matching method of object instances, which does not rely on the texture or shape descriptors instead of using geometric (*i.e.*, epipolar) constraint and a graph clustering. Experiments with simulated plant models demonstrated the proposed method yielded the average accuracy of the multi-view instance matching over 85%, which outperforms the performances of baseline methods based on descriptor-based approaches such as MVS. Our method also showed the potential to be used for instance-wise 3D reconstruction via the integration with 3D reconstruction methods such as the back projection.

Beyond the computer vision study, potential applications of the proposed method include the growth analysis of plants, as our experiments used a dataset from plant science and agricultural research field.

Acknowledgements. This work was supported in part by JST PRESTO Grant Number JPMJPR17O3.

References

1. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of International Conference on Computer Vision (ICCV). (2017)
2. Ren, M., Zemel, R.S.: End-to-end instance segmentation with recurrent attention. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 6656–6664
3. Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., Urtasun, R.: UPSNet: A unified panoptic segmentation network. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 8818–8826
4. Hou, J., Dai, A., Nießner, M.: 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 4421–4430
5. Kowdle, A., Sinha, S.N., Szeliski, R.: Multiple view object cosegmentation using appearance and stereo cues. In: Proceedings of European Conference on Computer Vision (ECCV). (2012) 789–803
6. Djelouah, A., Franco, J.S., Boyer, E., Le Clerc, F., Pérez, P.: Multi-view object segmentation in space and time. In: Proceedings of International Conference on Computer Vision (ICCV). (2013) 2640–2647
7. Mustafa, A., Hilton, A.: Semantically coherent co-segmentation and reconstruction of dynamic scenes. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 422–431
8. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B.: MOTS: Multi-object tracking and segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 7942–7951
9. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
10. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: Proceedings of European Conference on Computer Vision (ECCV). (2016)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal on Computer Vision (IJCV) **60** (2004) 91–110
12. Alcantarilla, P.F., Nuevo, J., Bartoli, A.: Fast explicit diffusion for accelerated features in nonlinear scale spaces. In: Proceedings of British Machine Vision Conference (BMVC). (2013)
13. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.R.: ORB: An efficient alternative to SIFT or SURF. In: Proceedings of International Conference on Computer Vision (ICCV). (2011)
14. Bleyer, M., Rhemann, C., Rother, C.: PatchMatch Stereo - stereo matching with slanted support windows. In: Proceedings of British Machine Vision Conference (BMVC). (2011)
15. Gallup, D., Frahm, J.M., Mordohai, P., Yang, Q., Pollefeys, M.: Real-time plane-sweeping stereo with multiple sweeping directions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2007)
16. Sun, J., Zheng, N.N., Shum, H.Y.: Stereo matching using belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **25** (2003) 787–800
17. Furukawa, Y., Sethi, A., Ponce, J., Kriegman, D.: Structure and motion from images of smooth textureless objects. In: Proceedings of European Conference on Computer Vision (ECCV). (2004) 287–298

18. Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., Liao, H.: Learning two-view correspondences and geometry using order-aware network. In: Proceedings of International Conference on Computer Vision (ICCV). (2019)
19. Romanoni, A., Matteucci, M.: TAPA-MVS: Textureless-aware patchmatch multi-view stereo. In: Proceedings of International Conference on Computer Vision (ICCV). (2019)
20. Nam, G., Wu, C., Kim, M.H., Sheikh, Y.: Strand-accurate multi-view hair capture. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 155–164
21. Dellaert, F., Seitz, S.M., Thorpe, C.E., Thrun, S.: Structure from motion without correspondence. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2. (2000) 557–564
22. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **24** (2002) 509–522
23. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 1. (2005) 26–33
24. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. International Journal on Computer Vision (IJCV) **65** (2005) 43–72
25. Bartolini, I., Ciaccia, P., Patella, M.: Warp: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **27** (2005) 142–147
26. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing **22** (2004) 761–767
27. Li, L., Zhang, S., Yu, X., Zhang, L.: PMSC: PatchMatch-based superpixel cut for accurate stereo matching. IEEE Transactions on Circuits and Systems for Video Technology **28** (2016) 679–692
28. Mičušík, B., Košecká, J.: Multi-view superpixel stereo in urban environments. International Journal on Computer Vision (IJCV) **89** (2010) 106–119
29. Klaus, A., Sormann, M., Karner, K.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: Proceedings of International Conference on Pattern Recognition (ICPR). Volume 3. (2006) 15–18
30. Kulikov, V., Lempitsky, V.: Instance segmentation of biological images using harmonic embeddings. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2020) 3843–3851
31. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Proceedings of Conference on Neural Information Processing Systems (NeurIPS). (2015) 91–99
32. Milan, A., Leal-Taixé, L., Schindler, K., Reid, I.: Joint tracking and segmentation of multiple targets. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 5397–5406
33. Jun Koh, Y., Kim, C.S.: CDTS: Collaborative detection, tracking, and segmentation for online multiple object segmentation in videos. In: Proceedings of International Conference on Computer Vision (ICCV). (2017) 3601–3609
34. Ošep, A., Mehner, W., Voigtlaender, P., Leibe, B.: Track, then decide: Category-agnostic vision-based multi-object tracking. In: Proceedings of International Conference on Robotics and Automation (ICRA). (2018)

35. Sharma, S., Ansari, J.A., Murthy, J.K., Krishna, K.M.: Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In: Proceedings of International Conference on Robotics and Automation (ICRA). (2018) 3508–3515
36. Seguin, G., Bojanowski, P., Lajugie, R., Laptev, I.: Instance-level video segmentation from object tracks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 3678–3687
37. Narita, G., Seno, T., Ishikawa, T., Kaji, Y.: PanopticFusion: Online volumetric semantic mapping at the level of stuff and things. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2019)
38. Schöler, F., Steinhage, V.: Automated 3D reconstruction of grape cluster architecture from sensor data for efficient phenotyping. Computers and Electronics in Agriculture **114** (2015) 163–177
39. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 652–660
40. Wang, L., Li, R., Sun, J., Liu, X., Zhao, L., Seah, H.S., Quah, C.K., Tandianus, B.: Multi-view fusion-based 3D object detection for robot indoor scene perception. Sensors **19** (2019) 4092
41. Lahoud, J., Ghanem, B., Pollefeys, M., Oswald, M.R.: 3D instance segmentation via multi-task metric learning. In: Proceedings of International Conference on Computer Vision (ICCV). (2019)
42. Dai, A., Nießner, M.: 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In: Proceedings of European Conference on Computer Vision (ECCV). (2018) 452–468
43. Hane, C., Zach, C., Cohen, A., Angst, R., Pollefeys, M.: Joint 3D scene reconstruction and class segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2013) 97–104
44. Savinov, N., Ladicky, L., Hane, C., Pollefeys, M.: Discrete optimization of ray potentials for semantic 3D reconstruction. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 5511–5518
45. Savinov, N., Hane, C., Ladicky, L., Pollefeys, M.: Semantic 3D reconstruction with continuous regularization and ray potentials using a visibility consistency constraint. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 5460–5469
46. Nassar, A.S., Lefèvre, S., Wegner, J.D.: Simultaneous multi-view instance detection with learned geometric soft-constraints. In: Proceedings of International Conference on Computer Vision (ICCV). (2019)
47. Wang, W., Yu, R., Huang, Q., Neumann, U.: SGPN: Similarity group proposal network for 3D point cloud instance segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 2569–2578
48. Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.J.: GSPN: Generative shape proposal network for 3D instance segmentation in point cloud. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 3947–3956
49. Engelmann, F., Bokeloh, M., Fathi, A., Leibe, B., Nießner, M.: 3D-MPA: Multi-proposal aggregation for 3D semantic instance segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2020) 9031–9040
50. Quan, L., Tan, P., Zeng, G., Yuan, L., Wang, J., Kang, S.B.: Image-based plant modeling. ACM Transactions on Graphics **25** (2006) 599–604

51. Kuang, D., Ding, C., Park, H.: Symmetric nonnegative matrix factorization for graph clustering. In: Proceedings of SIAM International Conference on Data Mining (SDM). (2012) 106–117
52. Girshick, R., Iandola, F., Darrell, T., Malik, J.: Deformable part models are convolutional neural networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 437–446
53. Deza, M.M., Deza, E.: Encyclopedia of Distances. Springer (2009)
54. Brooks, R.A., Di Chiro, G.: Theory of image reconstruction in computed tomography. Radiology **117** (1975) 561–572
55. Laurentini, A.: The visual hull concept for silhouette-based image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **16** (1994) 150–162
56. Isokane, T., Okura, F., Ide, A., Matsushita, Y., Yagi, Y.: Probabilistic plant modeling via multi-view image-to-image translation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
57. Tanabata, T., Hayashi, A., Kochi, N., Isobe, S.: Development of a semi-automatic 3D modeling system for phenotyping morphological traits in plants. In: Proceedings of Annual Conference of IEEE Industrial Electronics Society (IECON). (2018) 5439–5444
58. Catmull, E., Clark, J.: Recursively generated B-spline surfaces on arbitrary topological meshes. Computer-Aided Design **10** (1978) 350–355
59. Zhu, J., Du, S., Yuan, Z., Liu, Y., Ma, L.: Robust affine iterative closest point algorithm with bidirectional distance. IET Computer Vision **6** (2012) 252–261