Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2023) 31: 792 – 813

doi:10.55730/1300-0632.4018

Using t-distributed stochastic neighbor embedding for visualization and segmentation of 3D point clouds of plants

Helin DUTAĞACI*
Department of Electrical-Electronics Engineering, Faculty of Engineering and Architecture,
Osmangazi University, Eskişehir, Turkiye

Received: 13.02.2023 • **Accepted/Published Online:** 30.05.2023 • **Final Version:** 29.09.2023

Abstract: In this work, the use of t-SNE is proposed to embed 3D point clouds of plants into 2D space for plant characterization. It is demonstrated that t-SNE operates as a practical tool to flatten and visualize a complete 3D plant model in 2D space. The perplexity parameter of t-SNE allows 2D rendering of plant structures at various organizational levels. Aside from the promise of serving as a visualization tool for plant scientists, t-SNE also provides a gateway for processing 3D point clouds of plants using their embedded counterparts in 2D. In this paper, simple methods were proposed to perform semantic segmentation and instance segmentation via grouping the embedded 2D points. The evaluation of these methods on a public 3D plant data set conveys the potential of t-SNE for enabling 2D implementation of various steps involved in automatic 3D phenotyping pipelines.

Key words: Point clouds, plants, visualization, superpoint, segmentation, t-distributed stochastic neighbor embedding

1. Introduction

Automatic trait estimation of plants is becoming an indispensable component for many applications such as crop monitoring [1], crop quality assessment [2], agricultural robotics [3], and phenotyping [4, 5]. The developments in 3D reconstruction technologies and 3D computer vision tools allow 3D modeling of plants and their automatic trait estimation on digitized 3D models. Examples of such traits are leaf area, leaf length, leaf inclination angle, stem height, internode length and branching angle [6].

Characterization of 3D plants poses a particular challenge due to high structural heterogeneity, high self-occlusion, and high structural complexity. The complexity in the connectivity of parts, various parts occluding each other, and variation among parts cause difficulty for both manual and automatic inspection and delineation of the organs. Despite this challenge, 3D modeling and characterization of plants have been undertaken by the research community with considerable success [7]. Among the tasks that enable organ-based trait estimation are semantic segmentation and instance segmentation [8, 9]. In semantic segmentation, each point on the 3D surface is classified into one semantic category, such as leaf, branch, flower, or fruit. In instance segmentation, each point is assigned to a specific instance of an organ.

As a dimensionality reduction technique, t-distributed stochastic neighbor embedding (t-SNE) [10] is widely used for visualization of high-dimensional data in a low-dimensional space of two or three dimensions. It is a nonlinear dimensionality reduction method that aims to keep similar data points close in the lower-dimensional space and to preserve the local structure of the data. Traditionally, t-SNE is used only for visualization purposes.

*Correspondence: hdutagaci@ogu.edu.tr

However, in this paper, in addition to being a visualization tool for plant structures in various scales, t-SNE is demonstrated to be a suitable gateway for processing and clustering 3D point clouds in 2D space.

t-SNE has many capabilities suited to inspecting and processing 3D plant models in 2D space. One rationalization of the application of t-SNE to 3D plant data is its manifold assumption, which holds that the high-dimensional data points are inherently organized to lie on lower-dimensional manifolds [10]. This assumption resonates with the fact that, for certain plants species, much of the plant surface lies intrinsically on one-dimensional (thin branches, petioles) or two-dimensional (leaf blades, petals) manifolds. t-SNE is known to be capable of capturing the local geometry of high-dimensional data, thus unwrapping structures that lie on different low-dimensional manifolds. Thus, t-SNE makes it possible to visualize the complete unoccluded structure of the 3D plant via a single 2D view (Figure 1). Furthermore, t-SNE reveals both the local shapes of the parts and the global organization of the data in the form of clusters at multiple scales in the 2D space [10].

As stated, t-SNE produces a 2D point set organized in natural clusters and preserves local shape information that can be processed for information extraction. A brief review of t-SNE as a dimension reduction technique is given in Section 3.1. In this paper, three distinct applications of 2D embedding of 3D plant point clouds by t-SNE are proposed: 1) visualization of plant point clouds; 2) semantic segmentation of plant point clouds, and 3) instance segmentation of plant point clouds. Brief summaries of these applications are given as follows:

1. Plant visualization is a particularly important application for plant scientists and breeders. In Section 3.2, the utility of t-SNE for producing single shot views of 3D plant point clouds is demonstrated. These views allow the user to inspect various characteristics of a plant, gather information, and plan for further processing.

2. Semantic segmentation of 3D plant point clouds with the aid of t-SNE is discussed in Section 3.3. Semantic segmentation is performed through extraction of superpoints in the 2D space. The initial partitioning is based on the natural clusters formed by t-SNE. The clusters are then further partitioned through 2D shape characteristics and 2D spectral clustering in order to ensure convexity of the 2D segments. The final superpoint labels are then mapped to 3D and each superpoint is described by simple geometric features for classification.

3. Instance segmentation, in this work, is treated as leaf instance segmentation, where the objective is to identify, count, and separate individual leaves. In Section 3.4, a simple leaf instance segmentation algorithm through the use of t-SNE is introduced. Leaf points are embedded to 2D by t-SNE and partitioned into individual leaves through unsupervised 2D Euclidean clustering. Despite its simplicity, t-SNE proves to be effective to separate individual leaves, and yields instance segmentation performance comparable to the state-of-the-art methods.

The tomato point clouds of the Pheno4D data set [8] were used to test the proposed approaches. Key issues regarding plant visualization are discussed in Section 3.2. The quantitative performance results of semantic and instance segmentation methods are given in Section 4. The results demonstrate that with simple use of shape descriptors, 2D embedding by t-SNE gives comparable results to the state-of-the-art methods based on deep learning. To conclude, the contributions of this work are summarized as:

- To the best of the author's knowledge, this is the first attempt of using t-SNE for visualization of 3D point clouds of plants in 2D. t-SNE enables inspection of the parts of the entire 3D plant in a single 2D view.

- Similarly, to the best of the author's knowledge, 2D points embedded by t-SNE have not been processed for analysis of 3D point clouds. To this end, baseline methods for semantic and instance segmentation through the employment of t-SNE are developed.

- A novel method is proposed to organize 3D point clouds of plants in terms of superpoints via 2D processing of the points embedded by t-SNE.

2. Related work

This section is separated into two parts: In the first part, the use of t-SNE for data processing in computer vision is briefly given. In the second part, previous studies targeting semantic and instance segmentation of 3D plant models are discussed. Special emphasis is given to three aspects of the related work: 1) clustering approaches for segmentation of 3D plant models, 2) local surface features tailored to distinguish organ classes, 3) processing of 3D plant models in 2D space. In addition to these aspects, 3D point-based deep learning methods for plant analysis are mentioned, and the potential benefits of t-SNE in relation to deep learning approaches are introduced.

2.1. Data processing with t-SNE in computer vision

t-SNE is mostly used for visualization and understanding of high-dimensional data rather than generating a trainable model to process new data. It is extremely useful for interpretation of high-dimensional features generated by deep learning networks used in diverse types of computer vision problems. However, with the original t-SNE algorithm, it is not possible to use a t-SNE projection obtained from a training data set to process a new data set. It is mostly used as an unsupervised clustering technique applied to analyze individual data instances or data sets. Examples of such computer vision applications are analysis of intratumor heterogeneity from mass spectroscopy images [11], hyperspectral imaging analysis [12], and analyzing remote sensing datapoints to reveal differences between categories [13].

Although it is widely used for dimension reduction and clustering for interpretation and analysis of high-dimensional data, to the best of the author's knowledge, t-SNE has not been directly applied to 3D point clouds to process them in 2D domain. This work demonstrates that t-SNE allows segmentation of 3D plant point clouds in 2D domain due to the special structure of plant data.

2.2. Segmentation of 3D plant models

Clustering is a common approach to separating individual organs following semantic classification of 3D points on the plant surface. Individual leaves are either spatially detached from each other or exhibit strong geometric discontinuity at locations of contact, while structural units such as main stem, branches and petioles are connected. Thus, clustering is usually applied to regions containing points identified as leaves. For example, Elnashef et al. [14] clustered leaf points through DBSCAN to determine regions corresponding to individual leaves. To delineate single leaves in areas containing multiple overlapping leaves, Liu et al. [15] applied a version of DBSCAN that is based on manifold distance. Sun et al. [16] developed a density-based clustering method and applied it to 3D points identified as cotton bolls for the purpose of counting individual cotton bolls. DBSCAN can successfully separate leaves that have empty spaces or low-density regions between them. While DBSCAN is useful to group points that are closely packed together, it does not pay attention to the manifold structure

of the data as t-SNE does. t-SNE is sensitive to sharp changes in the underlying manifolds of the data, hence capable of separating touching leaves with different orientations.

Clustering can also be applied to the entire plant surface for an initial segmentation to be refined with further information. Golbach et al. [17] developed a segmentation method based on the breath-first flood-fill algorithm where close points are added to organ clusters iteratively. The flooding of an organ cluster stops whenever a measure of spread exceeds a threshold. Liu et al. [18] used Euclidean distance and spectral clustering (SC) algorithms for segmenting 3D point clouds into elementary shape units as candidate plant organs. Hétroy-Wheeler et al. [19] created a graph by using neighborhood information of 3D points of tree seedlings. Then, spectral clustering is applied to the graph for an initial partitioning. The procedure is repeated on the clusters to further segment them to leaves, petioles, and internodes. Mirande et al. [9] constructed a neighborhood graph to extract local geometric and spectral features for semantic segmentation. Then, they used a quotient graph to simultaneously refine the semantic labels and segment organ instances. t-SNE can be used as an alternative clustering technique for obtaining an initial segmentation for 3D plant data. It provides natural clusters in accord with the organization of the structural parts of the plant. Its main advantage over the other clustering techniques is its 2D output, where the local structure is preserved, and further refinement can be performed in the 2D space.

As in [16, 20, 21], an initial oversegmentation of the points can be obtained in the form of superpoints of a 3D point cloud. Li et al. [20] obtained an oversegmentation on plant point clouds to aggregate the segments into individual leaves via region growing. Duan et al. [21] partitioned point clouds using Octree organization into several groups, which corresponded to a slight oversegmentation of the cloud. The groups were then manually merged into individual organs. Sun et al. [16] used a voxel cloud connectivity segmentation (VCCS) algorithm to obtain an oversegmentation in the form of supervoxels. Supervoxels corresponding to bolls and branches of cotton plants were classified through SVM operating on region shape descriptors. Wang [22] organized the point cloud of a forest environment as a superpoint graph which is then processed by an unsupervised network architecture for both single tree isolation and leaf-wood classification. In their TreePartNet framework, Liu et al. [23] devised coupled networks that learned a neural decomposition of the input point cloud of a tree into a set of small-scale clusters, and merged oversegmented clusters through a pairwise affinity network. In this work, similarly, an oversegmentation of the 3D plant data in the form of superpoints is proposed. The advantage of the proposed method to the Octree organization is the delivery of natural partitioning of the plant data following the manifold structure of the plant organs. Furthermore, it is an unsupervised method eliminating the training stage as opposed to oversegmentation methods based on neural networks.

Clustering based on spatial proximity, connectivity, or point density provides a partitioning of the point cloud data; however, classification of the regions into semantic categories, such as leaf blades, stems, petioles, flowers, etc., requires descriptive information. Such descriptive information can be defined by quantifying the local geometric structure around the point to be classified. Dey et al. [24] computed point-ness, curve-ness, and surface-ness features from the eigenvalues of the local covariance matrix, and used SVM to classify the organs of grapevines. For leaf-stem classification, Li et al. [25] applied Markov Random Fields on the neighborhood graph of points. The data and smoothness terms were defined based on curvature values calculated using the eigenvalues from local PCA analysis. Elnashef et al. [14] extracted descriptors based on the first and

second tensor locally calculated around each point for semantic classification of leaves and stems. Using a deep learning classifier Ziamtsov and Navlakha [26] first selected Fast Point Feature Histograms (FPFH) as the best performing local feature among various alternatives. Then, they evaluated deep learning, support vector machines (SVM), K-nearest neighbors (KNN), and random forest on FPFH as classifiers. For simplicity, and for demonstrating the baseline success of the t-SNE-based superpoint extraction algorithm, a similar approach to the classical techniques reviewed above is used in this work for describing 3D plant points and superpoints.

Processing of 3D point clouds in 2D domain is another practice for plant segmentation, mainly due to the ease of applying powerful convolutional neural networks on data defined on regular grids. For point clouds reconstructed from multiple 2D color and/or depth images, 2D plant scenes are already available. For example, Shi et al. [27] exploited the 2D image data acquired for multiview 3D reconstruction of seedlings. They applied convolutional neural networks (CNN) to 2D images for semantic and instance segmentation. The predictions on 2D images were then combined through a voting strategy to label corresponding 3D points. Through an embedding network, Liu et al. [15] extracted features from depth images of poplar seedling leaves acquired from an RGB-D camera. Then, they applied Mask R-CNN to RGB-D data to determine leaf areas. Majeed et al. [28] developed a CNN-based segmentation network to segment foreground RGB images of apple trees into trunk, branch, and trellis-wire categories. The foreground images were extracted with the help of 3D point cloud data acquired with a Kinect V2 sensor. Although powerful, these methods are only applicable to 3D plant models that are reconstructed through 2D data such as RGB or RGB-D images. For 3D plant point clouds acquired by laser scanners, 2D representations should be virtually produced.

For point clouds for which there is no associated natural 2D representations, for example for LiDAR data, 2D rendering is performed virtually. In order to accomplish semantic labeling of 3D point clouds of grapevines, Japes et al. [29] generated 2D images, together with their associated labels, from multiple viewpoints around mesh representations of the clouds. They used these images to train a U-Net-based neural network. During inference, the prediction scores from 2D images were back projected to 3D. Itakura and Hosoi [30] converted the 3D plant model to volumetric form and projected it onto the plane above the model. The resulting binary image was segmented with watershed algorithm. Then, the partitions were shrunk to determine seed regions. The expansion of seed regions is performed in the 3D space to extract final leaf instances. Jin et al. [31] extracted horizontal slices from 3D point clouds of maize sites and the slices were projected into images as viewed from various angles. A faster R-CNN was trained on these images to detect individual maize plants. For crown segmentation of individual rubber trees from LiDAR data, Wang et al. [32] applied voxelization to the point cloud, and generated frontal and lateral projections of point clouds guided by the voxelization. These images were used to train a faster R-CNN that learned to locate individual tree trunks. For these methods, a number of issues should be addressed, such as the rendering method, number of images to be rendered, the resolution of the images, and the angles of the projection. With t-SNE, the entire 3D plant can be mapped onto a single 2D view without considering the pose of the 3D model or the locations and angles of the viewpoints.

In the current work, the use of t-SNE is suggested as a gateway for application of many of the methods mentioned above. The embedded 2D points and their natural clusters provided by t-SNE enable the implementation of 2D counterparts of those 3D techniques, such as superpoint extraction, split-and-merge approaches, graph-based methods, etc. Also, the clusters extracted by t-SNE can be encoded and labeled by well-studied

2D or 3D shape descriptors. This work includes the application of superpoint extraction in the 2D embedded space and description of them through local geometric features for the purpose of semantic segmentation of 3D plant models.

For the sake of completeness, deep learning approaches that aim to segment 3D plant point clouds are also mentioned in this section. Such techniques necessitate large amount of annotated data for training [33]. As 3D plant data sets are created [8, 34–36] and 3D annotation tools are developed [37, 38], application of 3D deep learning architectures to plant data is becoming feasable [8, 39]. General purpose 3D point-based architectures such as PointNet [40] and PointNet++ [41] were modified and applied to process 3D point clouds of plants [42–45]. Also, new architectures were developed specific to plant data such as RoseSegNet [46], plant segmentation transformer (PST) [47], Pattern-Net [48], PlantNet [49], PSegNet [50], DeepSeg3DMaize [51], and PartNet [52]. Such novel deep learning architectures will proliferate for plant phenotyping due to the power of deep neural networks. 2D embedding by t-SNE followed by preprocessing in 2D space has the potential of contributing to these networks for a number of reasons: 1) Partitioning large plant point clouds will be facilitated by the ability of t-SNE forming natural clusters; 2) Processing by point-based deep learning networks in 2D will require fewer parameters hence fewer training data; 3) The conversion of 2D point clouds to 2D binary images is less costly than 3D voxelization techniques, allowing efficient use of 2D convolutional neural networks. Another potential integration of the proposed superpoint extraction method with deep learning methods can be achieved via classifying the corresponding 2D/3D superpoints with deep neural networks.

3. Applications of t-SNE for 3D plant point cloud characterization

In this section, t-SNE is briefly described, and three distinct applications of 2D embedding of 3D plant point clouds by t-SNE are introduced: 1) Visualization of plant point clouds; 2) Semantic segmentation of plant point clouds, and 3) Instance segmentation of plant point clouds. Brief information about the Pheno4D data set [8], on which semantic and instance segmentation methods are evaluated, is also provided.

3.1. t-distributed stochastic neighbor embedding

t-distributed stochastic neighbor embedding (t-SNE) is a visualization technique that reorganizes high-dimensional data in 2D or 3D space [10]. To achieve a global organization in the form of clusters, t-SNE explores the implicit structure of all the data via random walks on neighborhood graphs.

In this paper, t-SNE algorithm, as introduced in [10], is utilized for embedding of 3D plant point clouds in 2D. Given a 3D point set $\mathcal{X}$ of a plant model, the specific objective of this study is to map the points in set $\mathcal{X}$ to a set $\mathcal{Y}$ of 2D points. To embed the points in $\mathcal{X}$ into 2D space, t-SNE attempts to minimize the Kullback-Leibler divergence between a Gaussian distribution of the points in the original space and a Student t-distribution of points in the target space. Kullback-Leibler divergence $KL(P \,||\, Q)$ between the joint probability distribution, P, in the high-dimensional space and the joint probability distribution, Q, in the low-dimensional space is given as:

$$KL(P \,||\, Q) = \sum_{j} \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{1}$$

where p_{ij} and q_{ij} correspond to joint probabilities of point pair (i,j) under the distributions P and Q, respectively.

The distance $d(x_i, x_j)$ between each pair of points x_i and x_j in the set $\mathcal{X}$ is calculated as a measure of similarity between the points. In this work, Euclidean distance is selected as the similarity measure. This similarity is used to estimate the conditional probability $p_{j|i}$, which can be interpreted as the probability of x_j occurring around x_i under a Gaussian distribution centered at x_i, and with variance σ_i. The conditional probability $p_{j|i}$ is estimated as:

$$p_{j|i} = \frac{exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-\|x_i - x_k\|^2/2\sigma_i^2)} \tag{2}$$

with $p_{i|i}$ set to 0. As stated, the conditional probability distribution, denoted as P_i, is assumed to be Gaussian and specific to the datapoint x_i. P_i embodies the distribution of all other data points in the set around x_i. t-SNE searches for the value of the variance σ_i such that the Shannon entropy $H(P_i)$ of P_i satisfies

$$Perp(P_i) = 2^{H(P_i)}. \tag{3}$$

The perplexity $Perp(P_i)$ is a user-defined parameter. The Shannon entropy of P_i is calculated as:

$$H(P_i) = -\sum_j p_{j|i} \log_2(p_{j|i}). \tag{4}$$

The joint probability p_{ij} is calculated by symmetrizing the conditional probabilities:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \tag{5}$$

where N is the number of points in set $\mathcal{X}$. The probability model q_{ij} of the distribution of the distances between points y_i and y_j in the low dimensional space is set to be a Student t-distribution with one degree of freedom. The joint probabilities q_{ij} are defined as:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \|y_k - y_l\|^2)^{-1}} \tag{6}$$

with q_{ii} set to 0. To minimize the Kullback-Leibler divergence between P and Q given in Eq. 1, t-SNE uses a gradient descent procedure. The details of this procedure can be found in [10]. t-SNE computes the conditional probability distribution P_i for each point x_i by setting a constant $Perplexity$ value. Thus, it adaptively adjusts the effective neighborhood size depending on local point density around each point [10].

3.2. t-SNE for visualization of plant point clouds

In this section, visualization of plant point clouds by t-SNE as the first proposed application is introduced. Visualization is one of the important services provided to biologists for inspection of the characteristics of the plant models under study. Visualization in 3D requires rotation of the model in 3D for allowing the viewer to

explore various structures from different views. t-SNE produces a complementary one-shot visualization where all components are flattened and laid out in 2D.

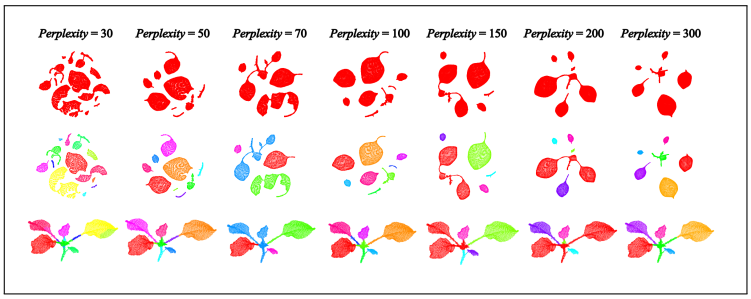

Figure 1. First row: t-SNE embeddings of a tobacco plant model from the Plant3D data set of Conn et al. [34] with varying *Perplexity*. Second row: t-SNE embeddings where the data is clustered through Euclidean clustering in 2D, and each cluster is rendered with a different color. Third row: The same 3D point cloud colored according to the clusters obtained from 2D embeddings.

Given a point cloud $\mathcal{X}$ composed of N 3D points, the point set $\mathcal{Y}$ including corresponding N 2D points are obtained through t-SNE as described in [10]. The parameters involved in the operation are the *Perplexity* and the number of points N, which is determined by the size of the plant and the point density. Depending on both parameters, the 2D embedding presents various layouts of the structures of the plant.

As stated before, t-SNE assumes that the high-dimensional data set is closer to a lower-dimensional manifold. It attempts to probe such manifold structures in the data and organize them in clusters. Many plant species have geometric structures that inherently follow lower-dimensional manifolds. The most notable example is a broad leaf blade whose points spread on a 2D continuous and smooth manifold. Thus, t-SNE has the potential to separate the plant data at sharp discontinuities in terms of orientation and group points together that lie on the same manifold. Furthermore, t-SNE embeds the points in each cluster onto the 2D space such that close points stay together, resulting in a faithfully flattened view of the clustered structures.

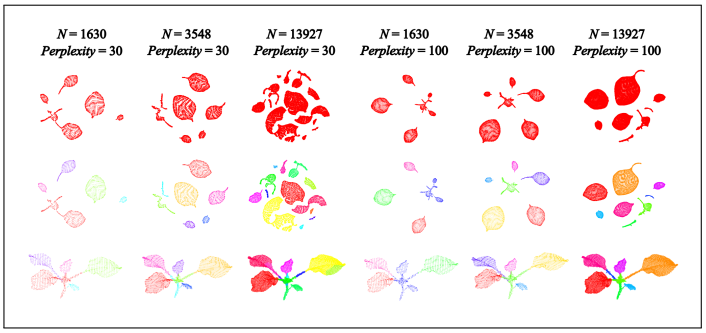

Figure 2. First row: t-SNE embeddings of a tobacco plant model from the Plant3D data set of Conn et al. [34] for three different resolutions and two *Perplexity* values. Second row: t-SNE embeddings where the data is clustered through Euclidean clustering in 2D, and each cluster is rendered with a different color. Third row: The same 3D point cloud colored according to the clusters obtained from 2D embeddings.

A tobacco plant with few organs and a simple architecture is chosen for the purpose of illustration of 3D plant visualization through t-SNE (Figures 1 and 2). Visualizations of more complicated plants are presented

in the Supplementary Material. The 3D model in Figures 1 and 2 belongs to the Plant3D data set described in [34]. The 3D point cloud is first subsampled to have $N = 13927$ points. Although the *Perplexity* parameter plays a significant role in the forming of clusters, a large range of *Perplexity* values (between 70 and 300 in the example shown in Figure 1) produces visually satisfactory embeddings that form natural clusters of the plant and reveal unrolled 2D shapes of the structures. Thus, once the desired level of organization is obtained, a fine tuning of the *Perplexity* value is not required.

At this level of point density ($N = 13927$), when *Perplexity* is 30, t-SNE tends to segment individual leaves at orientation discontinuities. When *Perplexity* is lower than 100, large leaves can be cut at orientation discontinuities. When the *Perplexity* is 100 and above, each individual leaf is entirely encompassed by a single cluster. The boundaries of the embedded leaves are in accordance with the corresponding 3D boundaries of leaves. This is a significant strength of t-SNE since it allows to render the shape of each leaf close to the shape if the leaf was flattened physically. Even small leaves that are barely visible in 3D and that require careful exploring to be noticed become apparent in 2D t-SNE embedding. From Figure 1, it can be observed that the small leaf at the branching point is observable as a small blob in the shape of a small leaf in all the 2D renderings. Through a single view, the observer can count the leaves and assess their shapes.

Another observation is that, with increasing *Perplexity* value, the 2D embedding reveals more about the plant architecture. Connected leaves start to stay connected in 2D through their corresponding petioles. Although not particularly significant for this example, this property of t-SNE allows grouping and rendering of substructures in 3D plant models with complicated architectures. Examples demonstrating this utility of t-SNE can be found in the Supplementary Material.

To observe the effect of point density, as quantified with number of points, N, in the point cloud, 2D embeddings of the same tobacco plant with three different subsampling rates are given in Figure 2. The tradeoff between efficient shape summarization and the retaining of detail is common to all 3D point cloud processing methods. The main disadvantage of low point resolution is the loss of discerning data for small structures. For example, although still visible, the small leaf at the branching point loses its discriminating shape in the embeddings with $N = 1630$ and $N = 3548$. However, when the number of points covering individual leaves drop, the tendency of t-SNE to partition individual leaves is no longer present at low *Perplexity* values. With low resolution, the leaf shape and the natural clusters are still observed. The branch structure appears as more connected in the 2D embeddings as compared to those from high resolution. As the resolution increases, individual branches and petioles appear as separate clusters.

3.3. Semantic segmentation

In this section, semantic segmentation as the second proposed application of t-SNE for 3D plant analysis is introduced. The flowchart of the proposed method is given in Figure 3. The objective is to assign to each point in the cloud one of the semantic labels: "Leaf" or "Stem". In the first stage, the point cloud is oversegmented into homogeneous regions; a procedure which is named as superpoint extraction. Superpoint extraction is performed in the 2D space where the 3D plant cloud is embedded by t-SNE. The details of this first stage are given in Section 3.3.1. The following stages involve classifying each superpoint as "Leaf" or "Stem". Local surface features are computed as descriptors of the superpoints and these descriptors are classified through a

support vector machine (SVM) classifier. The details of the descriptors are given in Section 3.3.2. Finally, a 3D point is assigned the semantic label of the superpoint containing it.

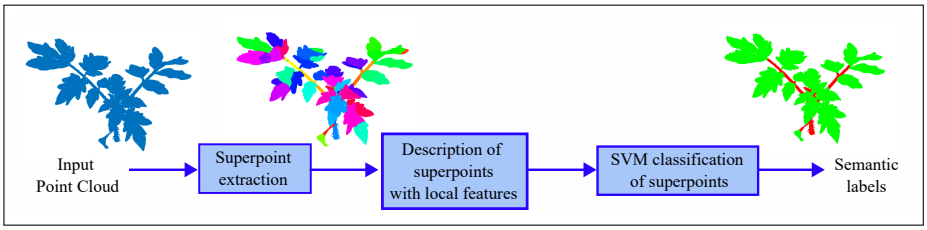

Figure 3. The flowchart for semantic segmentation. The input point cloud is partitioned into superpoints via the proposed algorithm based on t-SNE. Each superpoint is described via local surface features and then classified into semantic classes by SVM. The tomato model is from the Pheno4D data set of Schunck et al. [8].

3.3.1. t-SNE for superpoint extraction from 3D point clouds

As discussed in Section 2.2, segmentation approaches following a split-and-merge spirit are quite common for processing 3D point clouds of plants. Oversegmentation in the form of superpoints, which correspond to groups of points with similar local geometry, simplifies subsequent processing via treating the superpoint as a single and homogeneous unit [53]. At the first stage of the proposed semantic segmentation method, superpoints are extracted from 3D plant point clouds. To this end, a novel procedure that exploits t-SNE and generates a partitioning of the point cloud in the form of superpoints is proposed. The effectiveness of the proposed method is a demonstration of potential applications of t-SNE for 2D processing of 3D plant point clouds. The procedure starts with the clusters formed by t-SNE, then partitions each cluster further according to a number of criteria. Once the 2D superpoints are extracted, the superpoint labels are mapped to the 3D cloud.

The procedure of extraction of superpoints from the 2D embedding yielded by t-SNE is given in Figure 4. The input point cloud is downsampled via voxel grid average filtering before t-SNE is applied. The size of the voxel cube in the grid determines the point density and is denoted as v. Throughout this study, v is set to be 1 to ensure adequate point density to represent small leaves and, at the same time, low computational demand. If the number of points in the downsampled set is less than 1000, then v is decreased so that the number of points is close to 1000. This condition allows t-SNE to appropriately process small seedlings. Once the downsampled set of 3D points are mapped to 2D via t-SNE, the rest of the algorithm operates entirely in the 2D space until the superpoint labels are determined. The steps of superpoint extraction in 2D are summarized in Figure 4 with demonstrative examples. Algorithm 1 gives the implementation of 2D superpoint extraction. The steps are detailed as follows:

1. **2D Euclidean clustering:** First step in 2D is to apply Euclidean clustering to the embedded points generated by t-SNE (Line 1 in Algorithm 1). Euclidean clustering produces clusters such that, between any two points in a cluster, there is a path that remains in the cluster, and the distance between any subsequent points in the path is smaller than a threshold [54]. This threshold is denoted as d_E, and is one of the adjustable parameters of the superpoint extraction pipeline.

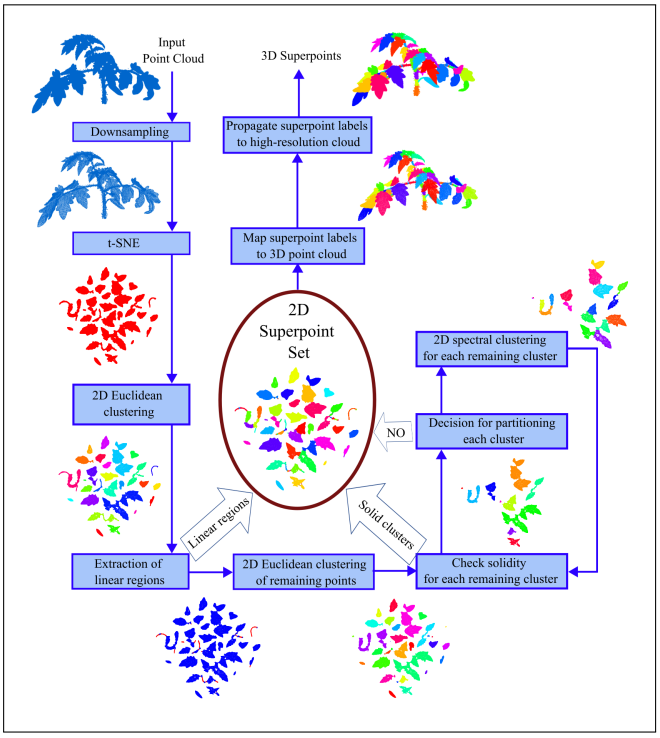

Figure 4. Procedure to extract superpoints from 2D points embedded through t-SNE. The tomato model is from the Pheno4D data set of Schunck et al. [8].

2. **Extraction of linear regions:** For each cluster obtained from the previous step, a local principal component analysis (PCA) is performed to extract thin, line-like regions. The objective here is to detect structures that might correspond to thin branches and petioles and separate them from other structures to which they are connected in the 2D embedding. For each point in each cluster, a local neighborhood of radius r_E is defined (Line 5 in Algorithm 1), and PCA is applied to the neighboring points to extract two principal directions and corresponding eigenvalues (Line 6 in Algorithm 1). r_E is always set to be equal to $2d_E$ throughout the experiments in this study. The ratio of the smaller eigenvalue to the larger is an indicator of the presence of a line-like structure at the locality. A small threshold $T_E = 0.1$ is chosen throughout all the experiments in order to extract points that belong to strictly line-like regions (Line 8 in Algorithm 1). The connected components of the points that have eigenvalue ratio smaller than T_E are formed through Euclidean clustering and these regions are included in the 2D Superpoint Set (Lines 9 and 10 in Algorithm 1).

3. **2D Euclidean clustering of remaining points:** After the removal of the line-like regions, 2D Euclidean clustering is applied to the remaining points, and the clusters are updated (Lines 11 and 12 in Algorithm 1). The structures connected by branches or petioles in the 2D embedding are thus disconnected through this operation.

4. **Check solidity for each remaining cluster:** For each remaining cluster that is not included in the Superpoint Set yet, the solidity is calculated (Line 15 in Algorithm 1). Solidity is a measure of the

convexity of a shape. In 2D, it is described as the ratio of the area of the shape to its convex area. The boundary points of each cluster are extracted and converted to a polygon. Through this polygon representation, the areas of the shape and its convex hull are calculated. The clusters with a solidity larger than $s_E = 0.8$ are deemed to be convex and added to the Superpoint Set (Line 17 in Algorithm 1). The remaining clusters are further examined for partitioning, since their concavity is an indicator that the cluster might include merged structures, such as merged individual leaves.

5. **Decision for partitioning each cluster:** For the remaining clusters whose solidity values are lower than s_E, spectral analysis is performed to decide whether or not to partition the cluster further. Spectral clustering is a graph-based clustering method that splits the graph in K_s partitions using the K_s eigenvectors of the Laplacian matrix of the graph [55, 56]. The number of clusters K_s is usually assumed to be equal to the number of eigenvalues of the Laplacian matrix that are close to 0. The algorithm as described in [56] is applied to the 2D points in a cluster, and K_s is estimated to be equal to the number of eigenvalues that are close to zero (Line 20 in Algorithm 1). The criterion to be close to zero is chosen to be being smaller than $T_s = 0.0005$ throughout the experiments. If K_s is equal to 1, indicating that the points in the cluster forms a single cluster, it is decided that no further partitioning is needed, and the cluster is added to the Superpoint Set (Line 22 in Algorithm 1).

6. **2D spectral clustering for each remaining cluster:** The clusters for which K_s is greater than 1 are partitioned into K_s subclusters through spectral clustering (Line 25 in Algorithm 1). The cluster is replaced by the subclusters. The set of remaining clusters, which are not yet assigned to superpoint labels, are thus updated (Lines 26 and 27 in Algorithm 1).

7. **Iterate through Steps 4 to 7:** The updated set of clusters are reprocessed through Steps 4 to 7 as long as the set of unlabeled clusters is nonempty.

An example of the resulting set of 2D superpoints is given in Figure 4, where each superpoint is rendered with a different color. The superpoint labels are mapped to the corresponding points in 3D, then propagated to the high-resolution point cloud through nearest neighbor interpolation. Most of the superpoints are entirely included in a single organ, representing a single leaf, part of a single leaf, or part of a branch. As a consequence, subsequent segmentation algorithms can operate on the superpoints treating each as an undivided, homogeneous entity.

3.3.2. Local surface features as descriptors of the superpoints

After the first stage of semantic segmentation procedure is completed by extracting superpoints, each superpoint is treated as a unity and represented by a single feature vector. Notice that a wide variety of descriptors can be used for representing the 2D or 3D point set of each superpoint, including deep features. In this study, geometric properties of spherical neighborhoods of 3D points were selected as baseline descriptors [57].

Let the original point cloud be described as a set of 3D points $\mathcal{X}$, where a point $x_i \in \mathbb{R}^3$ is represented with its 3D coordinates. A neighborhood of radius r is defined around each point x_i as $\mathcal{N}_i^r = \{x \in \mathcal{X} | \|x - x_i\| < r\}$,

where $\|\cdot\|$ is the Euclidean norm. Let the eigenvalues of the covariance matrix C_i^r of the 3D points in $\mathcal{N}_i^r$ be denoted as $\lambda_1 > \lambda_2 > \lambda_3$. Using these eigenvalues, the following shape descriptors of the local neighborhood $\mathcal{N}_i^r$ can be defined [57]: 1) Anisotropy: $f_{i,1}^r = \frac{\lambda_1 - \lambda_3}{\lambda_1}$, 2) Planarity: $f_{i,2}^r = \frac{\lambda_2 - \lambda_3}{\lambda_1}$, 3) Sphericity: $f_{i,3}^r = \frac{\lambda_3}{\lambda_1}$.

In order to take into account various sizes of neighborhoods, these shape descriptors are extracted setting the radius to 3 different values, $r = 4, 5, 6$. The radius values are selected such that they cover branches of various thickness. For each point x_i, in the set $\mathcal{X}$, a feature vector F_i of dimension 9 is calculated.

Let the set of superpoints be denoted as $\mathcal{SP}$, where superpoint $\mathcal{S}_n \in \mathcal{SP}$ is a subset of $\mathcal{X}$. All the 3D points in a single superpoint are supposed to belong to the same semantic category. To describe the superpoint $\mathcal{S}_n$, the mean of the feature vectors of the points in $\mathcal{S}_n$ is calculated:

$$FS_n = \frac{1}{N_n} \sum_{x_i \in \mathcal{S}_n} F_i, \tag{7}$$

where N_n is the number of 3D points in superpoint $\mathcal{S}_n$. The feature vector of each superpoint FS_n, is then used to classify the superpoint into one of two categories: "Leaf" and "Stem". A linear support vector machine (SVM) classifier is trained on a training set of plant models. For the training plants, the label of each superpoint is assigned as the majority ground truth category of the 3D points in the superpoint. At inference stage, after the label of a superpoint is predicted by SVM, all the 3D points in the superpoint are assigned to the same label.

Algorithm 1: Superpoint extraction in 2D space

Data: $\mathcal{Y}$: Set of 2D points embedded by t-SNE;
Result: $\mathcal{SP}_\mathcal{Y}$: 2D Superpoint set
Apply 2D Euclidean clustering to $\mathcal{Y}$ with d_E to get the set of clusters $\mathcal{C}$;
Initialize the set of line-like points as $\mathcal{B} \leftarrow \emptyset$;
foreach *set of points corresponding to a cluster* $C_i \in \mathcal{C}$ **do**
  foreach *2D point* $q_j \in C_i$ **do**
    Collect the set of neighboring points of q_j: $\mathcal{R}_{ij} = \{q_k \in C_i | \|q_j - q_k\| < r_E\}$;
    Compute the eigenvalues of λ_1 and λ_2 (with $\lambda_1 > \lambda_2$) of the 2D covariance matrix of $\mathcal{R}_{ij}$;
    if $\frac{\lambda_2}{\lambda_1} < 0.1$ **then**
      $\mathcal{B} \leftarrow \mathcal{B} \cup \{q_j\}$
Apply 2D Euclidean clustering to $\mathcal{B}$ with d_E to get the set of clusters $\mathcal{BC}$ of line-like regions ;
Initialize the superpoint set as $\mathcal{SP}_\mathcal{Y} \leftarrow \mathcal{BC}$;
Define the set of remaining points as $\mathcal{Y}_\mathcal{R} \leftarrow \mathcal{Y} \setminus \mathcal{B}$;
Apply 2D Euclidean clustering to $\mathcal{Y}_\mathcal{R}$ with d_E to get the set of remaining clusters $\mathcal{RC}$;
while $\mathcal{RC} \neq \emptyset$ **do**
  foreach *set of points corresponding to a cluster* $RC \in \mathcal{RC}$ **do**
    Compute the solidity s of the polygon formed by the boundary points of RC ;
    if $s > 0.8$ **then**
      $\mathcal{SP}_\mathcal{Y} \leftarrow \mathcal{SP}_\mathcal{Y} \cup \{RC\}$;
      $\mathcal{RC} \leftarrow \mathcal{RC} \setminus \{RC\}$;
    else
      Compute K_s of the points in RC using spectral analysis ;
      if $K_s = 1$ **then**
        $\mathcal{SP}_\mathcal{Y} \leftarrow \mathcal{SP}_\mathcal{Y} \cup \{RC\}$;
        $\mathcal{RC} \leftarrow \mathcal{RC} \setminus \{RC\}$;
      else
        Partition RC using spectral clustering into the set of clusters $\mathcal{C}_\mathcal{RC}$;
        $\mathcal{RC} \leftarrow \mathcal{RC} \setminus \{RC\}$;
        $\mathcal{RC} \leftarrow \mathcal{RC} \cup \mathcal{C}_\mathcal{RC}$;

3.4. Instance segmentation

In this section, instance segmentation as the third proposed application of t-SNE is described. The objective is to detect individual organs, e.g., individual leaves, and label each point with the identity of the organ that contains it. Instance segmentation can be performed on the entire unlabeled point cloud or proceed semantic segmentation. In the latter case, instance segmentation is applied only to the points labeled with a single category. 2D embedding by t-SNE allows for both strategies to be applied. Shape information is preserved in the 2D space to delineate individual plant organs through employing various leaf instance segmentation techniques that operate in 2D [58], including deep neural networks [59].

In this study, Euclidean clustering is applied to the embedded 2D points already classified as leaves by a semantic segmentation algorithm. The flowchart for instance segmentation via t-SNE is given in Figure 5. The objective is to demonstrate that, given the appropriate parameters for point resolution and $Perplexity$, the clusters formed by the unsupervised t-SNE algorithm significantly overlaps with individual leaves. Even with this naive approach based solely on t-SNE clusters, the performance of instance segmentation with t-SNE clusters is shown to be on par with that of a deep learning approach (see Section 4). The accuracy can be significantly improved by top-down (partitioning merged leaves) or bottom-up (merging torn leaves) refinement approaches without leaving the 2D domain.

Let a semantic segmentation algorithm label all 3D points in $\mathcal{X}$ with a set of labels including "Leaf". Let $\mathcal{L} \subset \mathcal{X}$ be the set of points labeled as "Leaf". First, $\mathcal{L}$ is downsampled using voxel grid average filtering with voxel size parameter v. In this work, v is selected to be equal to 1, to ensure that small leaves are discernible. Then, the downsampled set $\mathcal{L}_s$ is reduced to 2D via t-SNE algorithm resulting in the 2D point set $\mathcal{Y}_s$. Euclidean clustering [54] is applied to $\mathcal{Y}_s$ to obtain clusters L_k for $k = 1, 2, ..., K$ and label each point with its corresponding cluster index k. Each cluster index is assumed to correspond to a single leaf. The labels of 2D points are mapped to 3D counterparts in $\mathcal{L}_s$, then propagated via nearest-neighbor interpolation to the high-resolution point set $\mathcal{L}$.

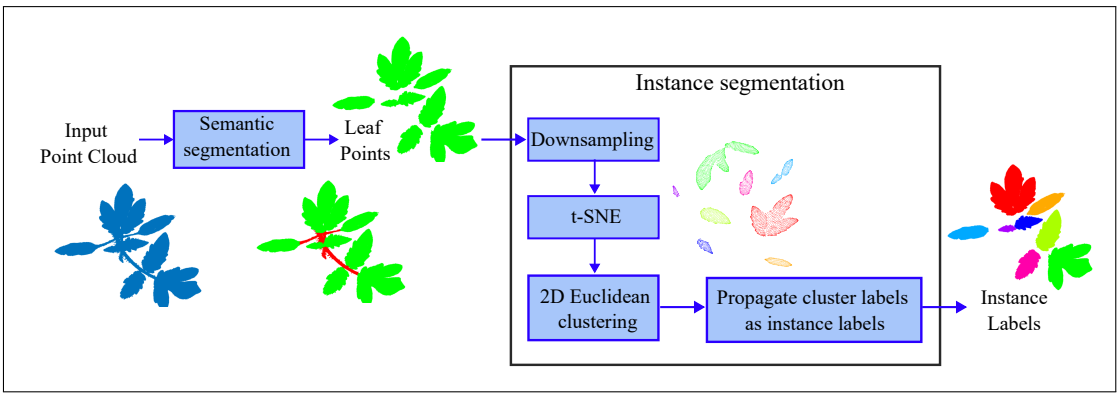

Figure 5. The flowchart for instance segmentation. Instance segmentation is applied only to the leaf points obtained by a semantic segmentation method. The tomato model in the figure is from the Pheno4D data set of Schunck et al. [8].

3.5. Data set

The publicly available Pheno4D data set [8] is chosen for evaluation of semantic and instance segmentation methods, since both semantic and instance ground-truth labels are provided in this data set. It contains point clouds of seven tomato plants acquired by a laser scanning system at various time instances. Eleven point clouds of each tomato plant were labeled, resulting in 77 labeled point clouds. The semantic labels correspond to "Leaf", "Stem", and "Ground". Instance labels are given as distinct identity numbers assigned to individual leaves.

4. Results

The experimental results of the semantic and instance segmentation methods based on t-SNE are reported in this section. The tests were performed on the 3D tomato point clouds of the Pheno4D data set [8].

4.1. Semantic segmentation

For semantic segmentation, intersection-over-union (IoU) per class and mean intersection-over-union (mIoU) are used as performance measures. The semantic labels provided in the Pheno4D data set are "Leaf", "Stem", and "Ground". For this study, the points corresponding to the "Ground" were manually removed since the focus of this work is to separate a plant into its organs. A preprocessing step, such as RANSAC can be used to detect the plane formed by the ground points. t-SNE also has the potential of separating ground points from the plant. In the Supplementary Material, visual results are given for the behavior of t-SNE when applied to the point clouds with the ground points.

Following the experimental setup described in [8], 55 point clouds of the first five plants were used for training, and the remaining two plants with 22 point clouds were reserved for testing. For the current study, the training set of plants was further partitioned into training and validation in a 4 to 1 ratio for searching best-performing $Perplexity$ value and the threshold d_E for Euclidean clustering. The values explored for $Perplexity$ were 20, 30, 40, and 50 through 5-fold validation experiments; resulting in 30 as the best performing $Perplexity$ value in terms of mIoU. For d_E, the search was done over the values 1, 2, and 3; with $d_E = 2$ giving the highest validation performance.

Table 1 gives the quantitative results of the proposed semantic segmentation procedure as compared to the results reported in [8]. The results were obtained with $Perplexity = 30$, and $d_E = 2$, since these were the best-performing parameters in the 5-fold validation experiments. Twenty-two point clouds of two plants were used as test data. Schunck et al. [8] had applied PointNet [40], PointNet++ [41], and LatticeNet [60] for semantic segmentation of the plant point clouds into three classes: "Leaf", "Stem", and "Ground". The current study excludes the classification of ground points. The mIoU values are given over three classes, as reported in [8], together with mIoU values over "Leaf" and "Stem" classes only.

Figure 6 gives visual examples for semantic segmentation on three point clouds acquired at different stages of growth of a tomato plant [8]. These clouds correspond to a small seedling, a medium-size tomato plant, and a larger plant with larger leaves. The respective size ratios are not kept in Figure 6 for clear illustration of the smaller plants. The first column (Figure 6a) shows the 3D point clouds with ground-truth instance labels of the leaves. In the second column (Figure 6b) the t-SNE embedding with $Perplexity = 30$ and

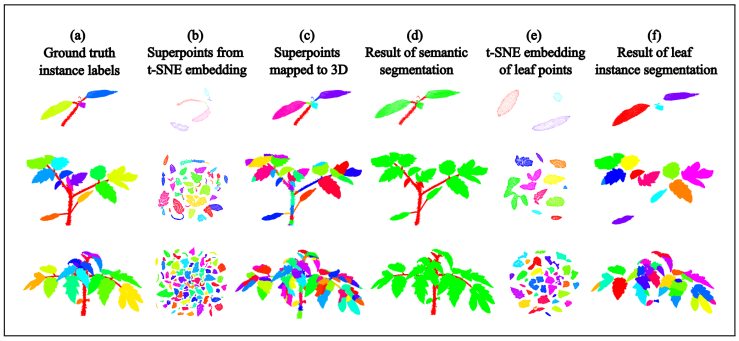

Figure 6. Visual examples for semantic and instance segmentation on three point clouds corresponding to different time instances of growth of a tomato plant. The models are from the Pheno4D data set of Schunck et al. [8].

Table 1. Semantic segmentation results given in IoU per class and mIoU on test tomato point clouds of Pheno4D data set [8]. Twenty-two point clouds of two plants were used as test data.

	IoU Leaf	IoU Stem	IoU Ground	mIoU over 3 classes	mIoU over 2 classes
PointNet [40]	83.9	11.9	98.4	64.7	47.9
PointNet++ [41]	88.8	19.6	98.6	68.4	54.2
LatticeNet [60]	98.7	86.9	99.8	95.1	92.8
Proposed	96.5	84.9	N/A	N/A	90.7

the extracted superpoints are given. For the seedling, small individual leaves remained in a single cluster, while for larger plants, large leaves were torn. However, the superpoint extraction algorithm managed to form large superpoints that include points from a single semantic class, and also discern small structures such as small leaves and petioles (Figure 6c). The results of the semantic segmentation in Figure 6d seem visually satisfactory apart from some small leaves classified as stem. Notice that these small leaves were successfully separated as superpoints; however, the SVM classifier working with local geometric features labeled them as stem. There is room for improvement, as future work, through testing other shape descriptors and classifiers, or deep learning schemes for representing and classifying superpoints more effectively.

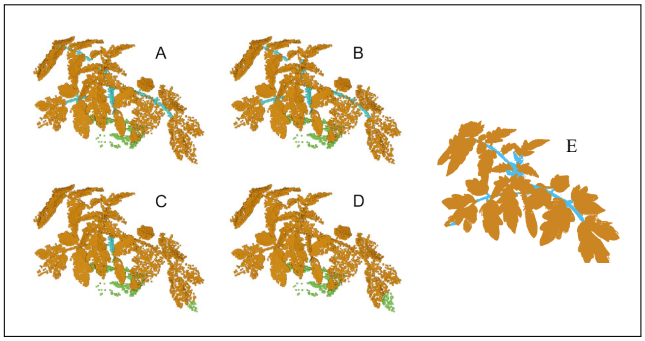

Figure 7. Visual examples for semantic segmentation provided by LatticeNet (B), PointNet++ (C), PointNet (D), and the proposed method (E). The ground truth is given in (A). The results depicted in (A), (B), (C), and (D) are from [8].

As can be observed from Table 1, the proposed method gives higher performance than PointNet and PointNet++, and is comparable to LatticeNet. In Figure 7, the segmentation results provided by LatticeNet (B), PointNet++ (C), PointNet (D), and the proposed method (E) are given for a sample tomato point cloud. PointNet++ misses the thin branches and PointNet is unable to detect stem points. LatticeNet gives the most successful segmentation followed by the proposed semantic segmentation method. The proposed method is able to separate thin stem regions from the leaf regions.

The results demonstrate that processing 2D points embedded by t-SNE has the potential of keeping up with state-of-the-art methods operating in 3D space for semantic segmentation of plants. Notice that the superpoint extraction is performed in an unsupervised manner, and the descriptors for the superpoints are hand-crafted and simple. Although powerful, deep learning methods such as PointNet++ and LatticeNet necessitate substantial amount of annotated training data. To be applicable to a different plant species, the networks need to be retrained with labeled 3D models of the new plant species. The proposed method, on the other hand, involves an unsupervised superpoint segmentation procedure, which reduces the problem to effective description of parts of leaves and stems. This final step can be indeed implemented by classical or deep learning-based machine learning techniques that target high generalization ability across a wide variety of plant species.

Additionally, as stated in Section 2.2, the use of 2D embedding by t-SNE followed by processing in 2D space does not exclude the employment of powerful deep learning approaches, such as LatticeNet. It would be worthwhile to apply LatticeNet, and other point-based deep learning architectures either for description of the superpoints or for segmentation of the 2D point clusters formed by t-SNE.

4.2. Instance segmentation

For evaluation of instance segmentation performance, symmetric best dice (SBD) [58] is used. In order to determine the best *Perplexity* value for the particular data set, instance segmentation as described in Section 3.4 is applied to the 55 point clouds of the first five tomato plants. The threshold for Euclidean clustering is set to $d_E = 2$. Points with ground-truth labeled as "Stem" are removed from the point clouds, and instance segmentation is applied only to the "Leaf" points. For determination of the *Perplexity* value for instance segmentation, perfect prior semantic segmentation is assumed to decouple errors from the semantic segmentation algorithm. Table 2 gives the SBD performances with respect to the *Perplexity* value. The highest SBD is yielded with *Perplexity* value equal to 60.

Table 2. Instance segmentation results on 55 point clouds corresponding to the first five tomato plants in the Pheno4D data set [8]. The performances with respect to *Perplexity* are given. Perfect prior semantic segmentation is assumed.

Perplexity	20	30	40	50	60	70	80
SBD	72.7	76.5	80.6	82.6	84.6	83.4	83.6

Table 3. Instance segmentation results given in SBD on 22 test tomato point clouds of Pheno4D data set [8].

	PointNet [40]	PointNet++ [41]	LatticeNet [60]	Proposed
SBD	47.3	56.1	74.2	73.5

In order to have a fair comparison with the instance segmentation results given in [8], the proposed instance segmentation method is tested on the 22 point clouds of the remaining two tomato plants. In this case, the proposed semantic segmentation is used as described in Section 3.3, and then, instance segmentation through t-SNE is applied to the points classified as "Leaf". *Perplexity* is set to be 60, as indicated by the results in Table 2. The SBD value of the proposed method is given in Table 3, together with the performance values as reported in [8]. With a simple unsupervised Euclidean clustering in the 2D space, and without further refinement, operating via t-SNE gives comparable instance segmentation performance (73.5%) to that of LatticeNet (74.2%).

Figures 6e and 6f give t-SNE embeddings of points classified as leaf, and the clusters provided by Euclidean clustering mapped to 3D. For small- and medium-size plants, the clusters mostly coincided entirely with a single leaf. For larger leaves, as shown in the third row, individual leaves with large concavities or with orientation discontinuities were torn into separate clusters. As a future work, strategies for merging these clusters through model-based techniques can be considered.

5. Conclusion

Through this work, t-SNE was discovered to be a powerful tool for visualization and processing of 3D point clouds of plants. It allows a single shot 2D view of the structures of the plant while preserving local shape information. Apart from visualization, t-SNE clusters plant data into natural partitions, leading to an initialization for successful segmentation methods. As a demonstration of this property, a novel superpoint extraction method that exclusively operates in 2D space is proposed. The effectiveness of this method is evaluated through semantic segmentation, where 90.7% mIoU is obtained on the Pheno4D tomato data set. The performance of t-SNE (73.5% SBD) via a simple unsupervised Euclidean clustering for instance segmentation is also shown to be on par with the state-of-the-art 3D deep learning techniques.

Acknowledgment

This study was supported by The Scientific and Technological Research Council of Türkiye (TÜBİTAK), Project No: 121E088.

References

[1] Yang G, Liu J, Zhao C, Li Z, Huang Y et al. Unmanned aerial vehicle remote sensing for field-based crop phenotyping: Current status and perspectives. Frontiers in Plant Science 2017; 8:1111. https://doi.org/10.3389/fpls.2017.01111

[2] Garbez M, Symoneaux R, Belin E, Caraglio Y, Chéné Y et al. Ornamental plants architectural characteristics in relation to visual sensory attributes: A new approach on the rose bush for objective evaluation of the visual quality. European Journal of Horticultural Science 2018; 83 (3): 187–201. https://doi.org/10.17660/eJHS.2018/83.3.8

[3] Roldán JJ, del Cerro J, Garzón-Ramos D, Garcia-Aunon P, Garzón M et al. Robots in agriculture: State of art and practical experiences. In: Neves AJR (editor). Service Robots. InTech, 2018, pp. 67–90. https://doi.org/10.5772/intechopen.69874

[4] Minervini M, Scharr H, Tsaftaris SA. Image analysis: The new bottleneck in plant phenotyping [applications corner]. IEEE Signal Prrocessing Magazine 2015; 32 (4): 126–131. https://doi.org/10.1109/MSP.2015.2405111

[5] James KMF, Sargent DJ, Whitehouse A, Cielniak G. High-throughput phenotyping for breeding targets—Current status and future directions of strawberry trait automation. Plants People Planet 2022. https://doi.org/10.1002/ppp3.10275

[6] Paulus S. Measuring crops in 3D: Using geometry for plant phenotyping. Plant Methods 2019; 15 (1): 1–13. https://doi.org/10.1186/s13007-019-0490-0

[7] 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

[8] Schunck D, Magistri F, Rosu RA, Cornelißen A, Chebrolu N et al. Pheno4D: A spatio-temporal dataset of maize and tomato plant point clouds for phenotyping and advanced plant analysis. Plos One 2021; 16 (8): e0256340. https://doi.org/10.1371/journal.pone.0256340

[9] Mirande K, Godin C, Tisserand M, Charlaix J, Besnard F et al. A graph-based approach for simultaneous semantic and instance segmentation of plant 3D point clouds. Frontiers in Plant Science 2022. https://doi.org/10.3389/fpls.2022.1012669

[10] Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research 2008; 9 (11).

[11] Abdelmoula WM, Balluff B, Englert S, Dijkstra J, Reinders MJ et al. Data-driven identification of prognostic tumor subpopulations using spatially mapped t-SNE of mass spectrometry imaging data. Proceedings of the National Academy of Sciences 2016; 113 (43):12244-9. https://doi.org/10.1073/pnas.1510227113

[12] Devassy BM, George S. Dimensionality reduction and visualisation of hyperspectral ink data using t-SNE. Forensic science international 2020; 311:110194. https://doi.org/10.1016/j.forsciint.2020.110194

[13] Song W, Wang L, Liu P, Choo KK. Improved t-SNE based manifold dimensional reduction for remote sensing data processing. Multimedia Tools and Applications 2019;78:4311-26. https://doi.org/10.1007/s11042-018-5715-0

[14] Elnashef B, Filin S, Lati RN. Tensor-based classification and segmentation of three-dimensional point clouds for organ-level plant phenotyping and growth analysis. Computers and Electronics in Agriculture 2019; 156:51–61. https://doi.org/10.1016/j.compag.2018.10.036

[15] Liu X, Hu C, Li P. Automatic segmentation of overlapped poplar seedling leaves combining mask R-CNN and DBSCAN. Computers and Electronics in Agriculture 2020; 178:105753. https://doi.org/10.1016/j.compag.2020.105753

[16] Sun S, Li C, Chee PW, Paterson AH, Jiang Y et al. Three-dimensional photogrammetric mapping of cotton bolls in situ based on point cloud segmentation and clustering. ISPRS Journal of Photogrammetry and Remote Sensing 2020; 160:195–207. https://doi.org/10.1016/j.isprsjprs.2019.12.011

[17] Golbach F, Kootstra G, Damjanovic S, Otten G, van de Zedde R. Validation of plant part measurements using a 3D reconstruction method suitable for high-throughput seedling phenotyping. Machine Vision and Applications 2016; 27:663–680. https://doi.org/10.1007/s00138-015-0727-5

[18] Liu J, Liu Y, Doonan J. Point cloud based iterative segmentation technique for 3D plant phenotyping. In: IEEE 2018 International Conference on Information and Automation (ICIA); 2018. pp. 1072–1077. IEEE 2018. https://doi.org/10.1109/ICInfA.2018.8812589

[19] Hétroy-Wheeler F, Casella E, Boltcheva D. Segmentation of tree seedling point clouds into elementary units. International Journal of Remote Sensing 2016; 37 (13): 2881–2907. https://doi.org/10.1080/01431161.2016.1190988

[20] Li D, Cao Y, Tang X-s, Yan S, Cai X. Leaf segmentation on dense plant point clouds with facet region growing. Sensors 2018; 18 (11): 3625. https://doi.org/10.3390/s18113625

[21] Duan T, Chapman S, Holland E, Rebetzke G, Guo Y et al. Dynamic quantification of canopy structure to characterize early plant vigour in wheat genotypes. Journal of Experimental Botany 2016; 67 (15): 4523–4534. https://doi.org/10.1093/jxb/erw227

[22] Wang D. Unsupervised semantic and instance segmentation of forest point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 2020; 165:86–97. https://doi.org/10.1016/j.isprsjprs.2020.04.020

[23] Liu Y, Guo J, Benes B, Deussen O, Zhang X et al. TreePartNet: Neural decomposition of point clouds for 3D tree reconstruction. ACM Transactions on Graphics 2021; 40 (6). https://doi.org/10.1145/3478513.3480486

[24] Dey D, Mummert L, Sukthankar R. Classification of plant structures from uncalibrated image sequences. In: IEEE 2012 Workshop on the Applications of Computer Vision (WACV); 2012. pp. 329–336. https://doi.org/10.1109/WACV.2012.6163017

[25] Li Y, Fan X, Mitra NJ, Chamovitz D, Cohen-Or D et al. Analyzing growing plants from 4D point cloud data. ACM Transactions on Graphics (TOG) 2013; 32 (6): 1–10. https://doi.org/10.1145/2508363.2508368

[26] Ziamtsov I, Navlakha S. Machine learning approaches to improve three basic plant phenotyping tasks using three-dimensional point clouds. Plant Physiology 2019; 181 (4): 1425–1440. https://doi.org/10.1104/pp.19.00524

[27] Shi W, van de Zedde R, Jiang H, Kootstra G. Plant-part segmentation using deep learning and multi-view vision. Biosystems Engineering 2019; 187:81 – 95. https://doi.org/10.1016/j.biosystemseng.2019.08.014

[28] Majeed Y, Zhang J, Zhang X, Fu L, Karkee M et al. Deep learning based segmentation for automated training of apple trees on trellis wires. Computers and Electronics in Agriculture; 2020 170:105277. https://doi.org/10.1016/j.compag.2020.105277

[29] Japes B, Mack J, Rist F, Herzog K, Töpfer R et al. Multi-view semantic labeling of 3D point clouds for automated plant phenotyping. arXiv preprint arXiv:1805.03994 2018.

[30] Itakura K, Hosoi F. Automatic leaf segmentation for estimating leaf area and leaf inclination angle in 3D plant images. Sensors 2018; 18 (10): 3576. https://doi.org/10.3390/s18103576

[31] Jin S, Su Y, Gao S, Wu F, Hu T et al. Deep learning: Individual maize segmentation from terrestrial lidar data using faster R-CNN and regional growth algorithms. Frontiers in Plant Science 2018; 9:866. https://doi.org/10.3389/fpls.2018.00866

[32] Wang J, Chen X, Cao L, An F, Chen B et al. Individual rubber tree segmentation based on ground-based LiDAR data and faster R-CNN of deep learning. Forests 2019; 10 (9): 793. https://doi.org/10.3390/f10090793

[33] Chaudhury A, Boudon F, Godin, C. 3D plant phenotyping: All you need is labelled point cloud data. In: European Conference on Computer Vision 2020; 244–260. Springer, Cham. https://doi.org/10.1007/978-3-030-65414-6_18

[34] Conn A, Pedmale UV, Chory J, Navlakha S. High-resolution laser scanning reveals plant architectures that reflect universal network design principles. Cell Systems 2017; 5 (1): 53–62. https://doi.org/10.1016/j.cels.2017.06.017

[35] Conn A, Pedmale, UV, Chory J, Stevens CF, Navlakha S. A statistical description of plant shoot architecture. Current biology 2017; 27 (14):2078–2088. https://doi.org/10.1016/j.cub.2017.06.009

[36] Dutagaci H, Rasti P, Galopin G, Rousseau D. ROSE-X: an annotated data set for evaluation of 3D plant organ segmentation methods. Plant methods 2020;16 (1):1-4. https://doi.org/10.1186/s13007-020-00573-w

[37] Miao T, Wen W, Li Y, Wu S, Zhu C, Guo, X. Label3DMaize: Toolkit for 3D point cloud data annotation of maize shoots. GigaScience 2021; 10 (5):giab031. https://doi.org/10.1093/gigascience/giab031

[38] Saeed F, Sun S, Snider J, Liu T, Li C. 3D annotation and deep learning for cotton plant part segmentatione and architectural trait extraction. 2022. PREPRINT available at Research Square. https://doi.org/10.21203/rs.3.rs-2179960/v1

[39] Turgut K, Dutagaci H, Galopin G, Rousseau D. Segmentation of structural parts of rosebush plants with 3D point-based deep learning methods. Plant Methods 2022; 18 (1):1–23. https://doi.org/10.1186/s13007-022-00857-3

[40] Qi CR, Su H, Mo K, Guibas LJ. PointNet: Deep learning on point sets for 3D classification and segmentation. In: IEEE 2017 Conference on Computer Vision and Pattern Recognition; 2017. pp. 652–660. https://doi.org/10.1109/CVPR.2017.16

[41] Qi CR, Yi L, Su H, Guibas LJ. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems 30 (NIPS); 2017; 30.

[42] Morel J, Bac A, Kanai T. Segmentation of unbalanced and in-homogeneous point clouds and its application to 3D scanned trees. The Visual Computer 2020; 36 (10):2419–2431. https://doi.org/10.1007/s00371-020-01966-7

[43] Chaudhury A, Hanappe P, Azaïs R, Godin C, Colliaux D. Transferring PointNet++ segmentation from virtual to real plants. In CVPPA-ICCV, 2021.

[44] Boogaard FP, van Henten EJ, Kootstra G. Boosting plant-part segmentation of cucumber plants by enriching incomplete 3D point clouds with spectral data. Biosystems Engineering 2021; 211:167–182. https://doi.org/10.1016/j.biosystemseng.2021.09.004

[45] Boogaard FP, van Henten EJ, Kootstra G. Improved point-cloud segmentation for plant phenotyping through class-dependent sampling of training data to battle class imbalance. Frontiers in Plant Science 2022; 13:838190–838190. https://doi.org/10.3389/fpls.2022.838190

[46] Turgut K, Dutagaci H, Rousseau D. RoseSegNet: An attention-based deep learning architecture for organ segmentation of plants. Biosystems Engineering 2022; 221:138–153. https://doi.org/10.1016/j.biosystemseng.2022.06.016

[47] Du R, Ma Z, Xie P, He Y, Cen H. PST: Plant segmentation transformer for 3D point clouds of rapeseed plants at the podding stage. ISPRS Journal of Photogrammetry and Remote Sensing 2023; 195:380-392. https://doi.org/10.1016/j.isprsjprs.2022.11.022

[48] Ghahremani M, Williams K, Corke FM, Tiddeman B, Liu Y, Doonan JH. Deep segmentation of point clouds of wheat. Frontiers in Plant Science 2021; 12:608732. https://doi.org/10.3389/fpls.2021.608732

[49] Li D, Shi G, Li J, Chen Y, Zhang S, Xiang S, Jin S. PlantNet: A dual-function point cloud segmentation network for multiple plant species. ISPRS Journal of Photogrammetry and Remote Sensing 2022; 184:243–263. https://doi.org/10.1016/j.isprsjprs.2022.01.007

[50] Li D, Li J, Xiang S, Pan A. PSegNet: Simultaneous semantic and instance segmentation for point clouds of plants. Plant Phenomics 2022. https://doi.org/10.34133/2022/9787643

[51] Li Y, Wen W, Miao T, Wu S, Yu Z et al. Automatic organ-level point cloud segmentation of maize shoots by integrating high-throughput data acquisition and deep learning. Computers and Electronics in Agriculture 2022; 193:106702. https://doi.org/10.1016/j.compag.2022.106702

[52] Wang L, Zheng L, Wang M. 3D point cloud instance segmentation of lettuce based on PartNet. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022. pp. 1647–1655. https://doi.org/10.1109/CVPRW56347.2022.00171

[53] Landrieu L, Simonovsky M. Large-scale point cloud semantic segmentation with superpoint graphs. In: IEEE 2018 Conference on Computer Vision and Pattern Recognition; 2018. pp. 4558–4567. https://doi.org/10.1109/CVPR.2018.00479

[54] Rusu RB. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. PhD, Technische Universitaet Muenchen, Munich, Germany, 2009.

[55] Shi J, Malik J. Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence 2000; 22 (8): 888–905. https://doi.org/10.1109/34.868688

[56] Ng A, Jordan M, Weiss Y. On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14 (NIPS); 2001; 14.

[57] Thomas H, Goulette F, Deschaud J-E, Marcotegui B, LeGall Y. Semantic classification of 3D point clouds with multiscale spherical neighborhoods. In: IEEE 2018 International Conference on 3D Vision (3DV); 2018. pp. 390–398. https://doi.org/10.1109/3DV.2018.00052

[58] Scharr H, Minervini M, French AP, Klukas C, Kramer DM et al. Leaf segmentation in plant phenotyping: a collation study. Machine Vision and Applications 2016; 27:585–606. https://doi.org/10.1007/s00138-015-0737-3

[59] Gu W, Bai S, Kong L. A review on 2D instance segmentation based on deep neural networks. Image and Vision Computing 2022; page 104401. https://doi.org/10.1016/j.imavis.2022.104401

[60] Rosu RA, Schütt P, Quenzel J, Behnke S. LatticeNet: Fast spatio-temporal point cloud segmentation using permutohedral lattices. Autonomous Robots 2022; 46 (1): 45–60. https://doi.org/10.1007/s10514-021-09998-1