Proximity-Based Label Propagation for Unstructured Point Clouds: The K-Neighbor-Nurtured-Garden Algorithm

Gunner Stone

Abstract—**The application of Terrestrial and Airborne Laser Scanning in forestry has demonstrated its effectiveness in generating digital representations of landscapes through point cloud datasets. Automated semantic labeling of individual trees, utilizing point cloud segmentation models, can yield valuable information on topological, geometrical, and volumetric aspects of tree structures. However, these semantic labeling techniques often depend on fixed-size inputs, potentially limiting the level of detail for certain forestry applications.**

This study introduces the K-Neighbor-Nurtured-Garden (KNNG) algorithm, a label propagation technique designed to address this limitation. The algorithm operates directly on unstructured point clouds and propagates labels based on proximity, without relying on a predefined graph structure. The performance of the KNNG algorithm will be evaluated against other community detection and clustering methods.

This research aims to advance the field by presenting a potentially more effective and versatile solution for holistic semantic labeling in forestry applications. Our results demonstrate the superiority of K-NN connectivity inference in combination with the most suitable community detection method for a given dataset. The outcomes of this study could enhance the accuracy and granularity of information obtained from Terrestrial and Airborne Laser Scanning, ultimately offering valuable insights for forestry professionals.

Index Terms—**L-Systems, TLS, ALS, Pointcloud, Remote Sensing, Tree Part Segmentation, Point Voxel Transformer, PVT.**

I. Introduction

FORESTS are one of the most important natural resources on Earth, providing essential ecological services, including carbon sequestration, biodiversity conservation, and water regulation, among others[1]. The study of forestry is critical to the management and preservation of these ecosystems. With the increasing availability of remote sensing technologies, such as Terrestrial and Airborne Laser Scanning (TLS and ALS), digital cataloging of landscapes in the form of pointcloud datasets has become feasible[2]. Such data can provide valuable information on the structure and composition of forest ecosystems at different scales, from individual trees to entire landscapes.

To extract meaningful information from these vast datasets, automated segmentation and labeling of the pointcloud data have become important research topics. Such techniques are used to identify different components of a tree, such as the trunk, branches, and leaves, and provide valuable information on the topological, geometrical, and volumetric details of the tree structure. This information can then be used to estimate important forest parameters, such as tree height, diameter at breast height, crown diameter, volume, and biomass.

However, automated segmentation and labeling techniques for point cloud data are often limited by the fixed-size inputs used in the segmentation models. These limitations may result in inaccurate or incomplete labeling, which can negatively impact the analysis of tree structures and forest parameters. Moreover, manual segmentation and labeling of the pointcloud data are time-consuming and subject to errors, making them unfeasible for large datasets.

To address these limitations, a new region growing technique has been proposed in this study to see if it can provide a more effective solution for automated semantic labeling in the context of forestry applications. The K-Neighbor-Nurtured-Garden (KNNG) algorithm is a novel graph-agnostic label propagation technique that assigns labels to nodes in a pointcloud based on their proximity to other nodes, referred to as ”flowers”. By iteratively identifying and propagating labels through ”seeds”, the algorithm adapts to various graph structures. The proposed technique can provide more accurate and detailed information on tree structures, compared to other conventional modularity maximization techniques, such as Spectral Lanczos and Fast Unfolding Modularity Maximization (FUMO).

In this paper, we aim to identify the most effective label propagation techniques for point cloud data, specifically for synthetic tree point clouds generated using Speedtree and semantically segmented using a Point-Voxel Transformer model. Our primary contributions include a comprehensive comparison of various community detection methods and graph connectivity inference techniques, as well as the proposal of a novel K-Neighbor-Nurtured-Garden (KNNG) method.

In this study, the performance of the proposed region growing technique will be evaluated and compared with other standard semi-labeled clustering methods such as Voronoi clustering and K-NN. The evaluation will consider the accuracy, efficiency, and effectiveness of the proposed technique for semantic labeling of trees.

II. Related Works

Recent advancements in point cloud segmentation have largely been driven by deep learning techniques. One notable approach is PointNet [3], which processes individual points and captures local information. This has been further improved by PointNet++ [4], which introduces a hierarchical neural network that is better able to capture local and global features. Other deep learning approaches include SpiderCNN [5], which introduces a spider convolution operator, and KPConv [6], which uses kernel point convolution to directly process points in space.

Another example is the work by Yang et al. [7], which proposes a framework for 3D object detection and instance segmentation in point clouds. Their approach uses a graph neural network to encode local and global features of the point cloud, and then performs object detection and segmentation by predicting bounding boxes and instance masks. They also use label propagation to refine object segmentation results, where labeled objects are used to guide the labeling of neighboring objects in the point cloud.

Recent works have shown that deep learning-based approaches are highly effective at semantic segmentation of tree point clouds. For example, Point-voxel transformer (PVT) [8] achieved state-of-the-art results on the task of segmenting trunk, branch, and leaf points. Additionally, VoxelNet [9] and SECOND [10] are two popular approaches that perform 3D object detection and segmentation by transforming point clouds into a voxel representation.

While deep learning-based techniques have demonstrated remarkable performance in the semantic segmentation of point clouds, a significant limitation of these methods is the fixed input size of the model. This means that the entire point cloud of a tree often needs to be heavily subsampled to fit into the model, resulting in a loss of information and potentially poorer segmentation outcomes. One approach to tackle this challenge is to use partially labeled point clouds from the outputs of these models to guide the clustering process. By using the labeled points as seeds or anchors to propagate labels to neighboring points using methods such as modularity maximization or graph-based techniques, the clustering process can take advantage of the partially labeled data, potentially enhancing the accuracy and efficiency of the method. The KNNG algorithm, introduced in this study, takes advantage of this by processing the entire, unaltered point cloud, without requiring subsampling, offering a more adaptable solution for semantic labeling via label propagation. These techniques are especially useful for forestry applications where subsampled point clouds may not provide enough detail about the topology of the trees.

Traditional unsupervised clustering approaches include Voronoi clustering, K-clustering, among others. DBSCAN [11] is a popular unsupervised clustering method that groups together points with high spatial density, effectively separating them from low-density points. Other graph-based methods include Random Walks and Efficient Graph-based Segmentation (EGS) [12], which use a graph representation to segment the point cloud.

Community detection is a fundamental problem in network analysis, and has been extensively studied in various domains including social networks, biological networks, and transportation networks. The goal of community detection is to identify densely connected groups of nodes in a network that share similar structural properties. These communities can be interpreted as clusters of nodes with high intra-cluster connectivity and low inter-cluster connectivity. In the context of point cloud analysis, these communities can be seen as groups of points with similar characteristics, such as spatial proximity, shape, or color. By partitioning the point cloud into communities, it becomes possible to identify and analyze structures or patterns that may not be immediately apparent from the raw data. For example, in forestry applications, communities may correspond to different parts of a tree, such as branches or leaves.

Modularity maximization is a widely used community detection technique that aims to identify the optimal partitioning of nodes in a network into densely connected communities by measuring the density of connections within communities compared to that between communities [13]. The modularity optimization problem is known to be NP-hard, but several efficient algorithms have been proposed to solve it, such as the Louvain algorithm [14], Infomap [15], and the spectral algorithm [16]. The Louvain algorithm is a fast and scalable algorithm that iteratively optimizes the modularity score of the network by moving nodes between communities. Infomap is a flow-based algorithm that seeks to find the partitioning that minimizes the description length of a random walk on the network. The spectral algorithm uses the eigenvectors of the modularity matrix to find the optimal partitioning of the network. These algorithms have been successfully applied in various fields, such as social networks, biology, and computer science, and can also be adapted for clustering partially segmented point clouds.

FUMO (Fast unfolding of communities in large networks) algorithm [17], an extension of the Louvain algorithm, has also been employed in community detection. FUMO is efficient and can handle larger networks. It has been used in various applications, including community detection in social networks [18], gene co-expression networks [19], and brain functional networks [20].

However, modularity maximization is not optimal for semantic labeling of trees in large pointcloud datasets due to its computationally intensive nature, which becomes increasingly problematic with large datasets. Moreover, the quality of the clustering results highly depends on the initial partition, which can be challenging to determine accurately for complex tree structures. Therefore, alternative clustering methods such as the proposed region growing technique should be explored to overcome these limitations and improve the accuracy and efficiency of semantic labeling in the context of forestry applications.

Label propagation algorithms are a popular approach to clustering in partially labeled data. Cordasco and Gargano[21] proposed a semi-synchronous label propagation algorithm (SSLP) for community detection in social networks. In their approach, each node is assigned a unique label initially, and a subset of nodes are designated as seeds with known labels. The algorithm then iteratively updates the labels of the unlabeled nodes using the labels of their neighboring nodes until convergence. The updates are semi-synchronous, meaning that the algorithm updates the labels of the nodes in batches to reduce the communication overhead. The authors showed that their approach is effective in detecting communities in real-world social networks and can outperform other popular community detection algorithms such as the Louvain algorithm and the Girvan-Newman algorithm in terms of accuracy and efficiency. The SSLP algorithm is promising for labeling partially segmented point clouds as well since it can use the labeled points as seeds and propagate the labels to the neighboring unlabeled points to generate clusters.

Fluid Communities is a highly scalable and competitive community detection algorithm proposed by Parés et al. in their paper [22]. The algorithm is based on the idea of fluid dynamics and simulates the flow of particles in a network. The nodes in the network are assigned to different communities based on the density and direction of particle flow. The algorithm is highly scalable and efficient, making it suitable for large-scale point cloud datasets. The authors demonstrated the effectiveness of Fluid Communities on a variety of real-world datasets, including social networks and biological networks, achieving state-of-the-art performance.

Overall, community detection provides a powerful framework for labeling partially segmented point clouds and identifying objects or segments within the data. The key challenge is to design effective features and similarity measures that capture the underlying structure of the data, and to develop efficient algorithms that can scale to large point clouds with millions of points. The use of label propagation, as shown in the work by Cordasco et al., can further refine object segmentation results by leveraging labeled objects to guide the labeling of neighboring objects in the point cloud.

III. Methodology

The K-Neighbor-Nurtured-Garden (KNNG) algorithm is a versatile label propagation algorithm that can be used to label nodes in an unstructured point cloud without relying on a pre-built graph structure. The algorithm consists of two main passes: a "seeds" creation pass and a "seeds" transformation pass. The details of these passes are illustrated in Figure 1. In the first pass, the algorithm identifies the K closest "dirt" nodes to each "flower" using an Annoy index for fast nearest neighbor search. These "dirt" nodes are then assigned the same label as the "flower", transforming them into "seeds". In the second pass, the "seeds" are transformed into new "flowers", which participate in the next iteration of the labeling process. The algorithm continues running until there are no more "dirt" nodes or "flowers" remaining.

The algorithmic implementation of the KNNG algorithm involves the following steps:

function K-Nearest-Nurtured-Garden(G,K)
  Initialize an Annoy index for fast nearest neighbor search.
  Get partially labeled nodes and assign them as "flowers" F.
  while there are "dirt" nodes D or "flowers" F remaining in G **do**
    for each "flower" $f \in F$ **do**
      Find the K closest "dirt" nodes $d \in D$ within the z-score threshold.
      Assign f's label to the "dirt" nodes, turning them into new "seeds" S.
    end for
    Transform the "seeds" S into new "flowers" F' for the next iteration.
    Update the set of "dead" nodes X to include any "flowers" that did not generate new "seeds" in the previous iteration.
    Remove "dead" nodes X from G.
  end while
  return the list of nodes with labels assigned by the KNNG algorithm.
end function

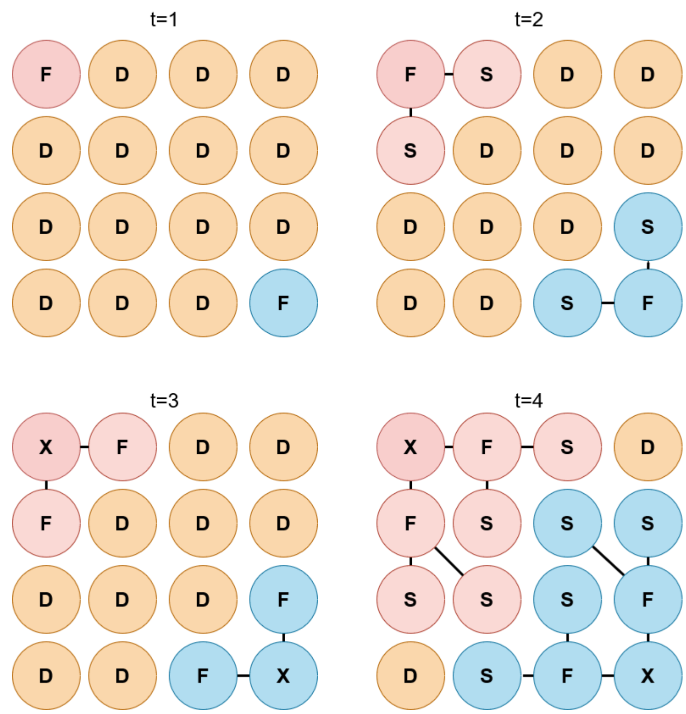

Fig. 1. Four time steps of the K-Neighbor-Nurtured-Garden (KNNG) algorithm.

The presented K-Neighbor-Nurtured-Garden (KNNG) algorithm was developed to efficiently propagate labels within an unstructured point cloud without relying on a predefined graph structure. This makes the KNNG algorithm more versatile and useful for applications such as tree segmentation, where a pre-built graph may not exist or may not accurately capture the structure of the point cloud. Unlike community detection techniques that require a pre-built graph or network, the KNNG algorithm works directly with unstructured point clouds and makes no assumptions about the input.

In the following sections, we will investigate the efficacy of various community detection techniques in our task, which requires the construction of a graph from the point cloud. To create such a graph, we consider two graph construction methods and evaluate their potential for label propagation in this context. After discussing the chosen graph construction techniques, we will delve into the limitations of other surface reconstruction methods and explain why they are not well-suited for reconstructing tree surfaces in our application.

To analyse various community detection techniques for their

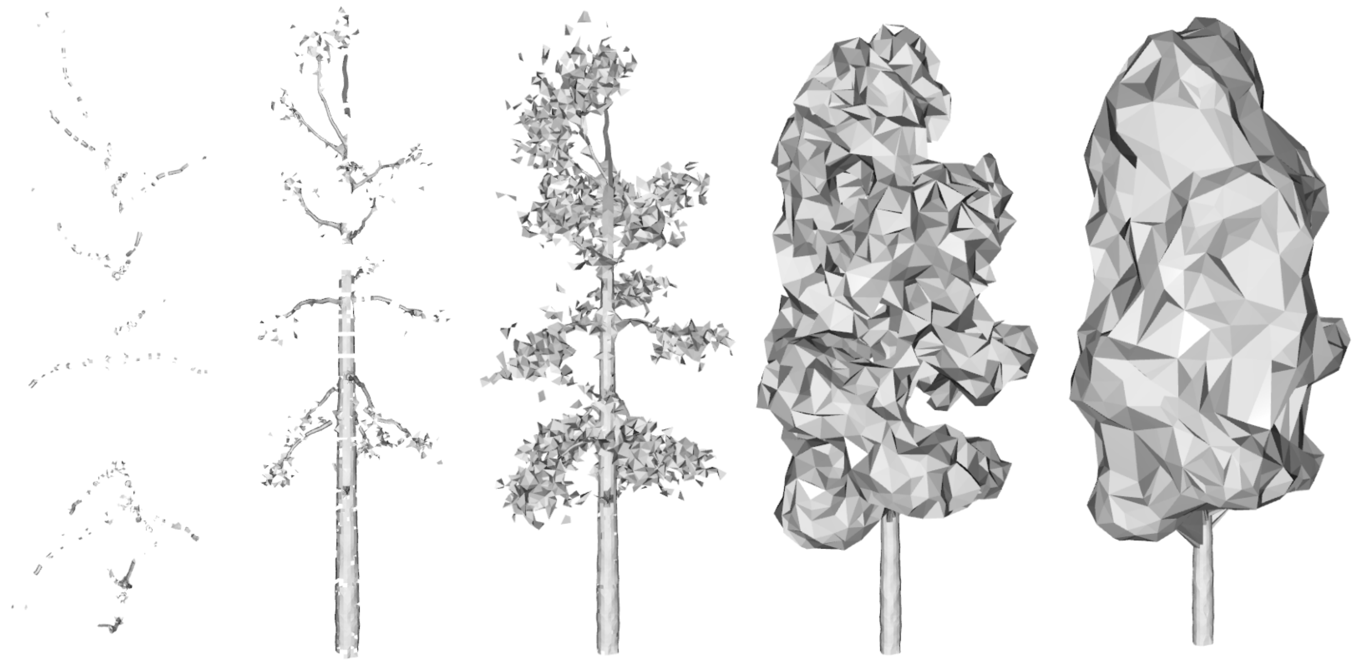

Fig. 2. Alpha-Shapes mesh generation for a tree pointcloud with increasing alpha values ($\alpha = 0.005$, $\alpha = 0.01$, $\alpha = 0.03$, $\alpha = 0.05$, $\alpha = 0.1$).

effectiveness in this task, it is necessary to construct a graph from the pointcloud. In this study, two graph construction methods were analyzed for their effectiveness in propagating labels. The first method involved naively constructing edges between a node and its K closest neighbors, with a low value of K chosen to give nodes a more local connectivity (K = 10 was used in this study). The second method utilized alpha shapes to approximate a mesh for the tree pointcloud, as shown in Figure 2. Alpha shapes[23] is a surface reconstruction technique that uses a parameter alpha to determine the coarseness of the surface reconstruction. The alpha shapes algorithm does not guarantee a single connected-component graph, which is necessary for certain community detection techniques. Because of this limitation, if more than one graph component was created, they were stitched together using the closest neighbors between components. Another limitation of this graph construction approach is that not all points from the original pointcloud are included in the alpha shapes resultant mesh. To address this, points that were not part of the resulting mesh became tethered to their closest neighboring mesh-node.

In the field of surface reconstruction, there are various techniques that have been proposed to generate surface models from point cloud data. Among these are methods such as Poisson Surface Reconstruction (PSR) [24], Ball Pivoting Algorithm (BPA) [25], and Marching Cubes (MC) [26]. However, these methods may not be suitable for reconstructing tree surfaces due to several reasons.

Poisson Surface Reconstruction (PSR) is a technique that requires normal vector information for the input point cloud [24]. This method reconstructs the surface by solving a Poisson equation on an octree data structure. However, in the case of tree point clouds, normal vector information may not always be readily available, rendering PSR infeasible for such applications.

The Ball Pivoting Algorithm (BPA) reconstructs surfaces by simulating the rolling of a ball over the point cloud data [25]. The algorithm starts with a seed triangle and iteratively extends the surface by adding new triangles that fit the ball. Although BPA can generate detailed surface models, it often struggles with complex and concave shapes, such as tree branches and foliage. Additionally, the algorithm is sensitive to the choice of ball radius, which can be challenging to determine accurately for irregular tree structures.

Marching Cubes (MC) is a widely used technique for extracting isosurfaces from volumetric data [26]. This method divides the input data into a grid of cubes and processes each cube independently to generate a surface. While MC can handle complex shapes, it requires the input data to be in a volumetric

Although these surface reconstruction techniques have shown success in various applications, they might not be well-suited for tree point clouds due to their requirements for specific data formats, normal vector information, or their inability to handle complex and concave shapes. In contrast, KNN connectivity inference and Alpha Shapes Surface Reconstruction offer more flexible and adaptable solutions for constructing graphs from point clouds, making them more suitable for tree surface reconstruction tasks.

Label propagation within clusters is a process of assigning labels to a set of unlabeled data points within a cluster based on a small set of labeled data points that belong to the same cluster. In this study, the goal is to use this approach to label a point cloud of a tree that has partially labeled data available. The first step is to identify clusters of points within the tree point cloud. This can be done using community detection techniques, which group nodes with similar attributes or connections into clusters. Once the clusters are identified, label propagation can be performed within each cluster separately. The idea is to use the labeled data points available in a cluster to propagate the label to the neighboring unlabeled points within that cluster. This can be done using techniques such as modularity maximization, graph-based approaches, or the K-Neighbor-Nurtured-Garden (KNNG) algorithm. The effectiveness of the labeling process within each cluster can be measured by comparing the predicted labels to the ground truth labels for the unlabeled points within that cluster.

To implement this approach, 1000 synthetic point clouds of trees with ground truth point-wise class labels were generated using Speedtree. A PointVoxel Transformer segmentation model was trained on a training/testing split of 90% and 10%, respectively, to accept subsampled point clouds of size 2048. The model's effectiveness was tested on the test set using both a 2048 subsample and no subsampling at all.

To propagate labels to the unlabeled points in the tree, the labeled 2048 points were used as inputs to various techniques: FUMO, Spectral Lanczos, AFC, Label Propagation, Async LPA, Recursive KNN, Voronoi Clustering, and KNNG. These techniques were chosen based on their relevance to the problem and their effectiveness in related works. The goal was to compare the performance of these techniques in label propagation within clusters and evaluate their accuracy and efficiency.

The FUMO technique is a graph-based clustering method that uses a Fast Unfolding algorithm to identify communities in a graph. Spectral Lanczos is a spectral clustering algorithm that uses the Lanczos algorithm to approximate the eigenvectors of the graph Laplacian matrix. AFC is a graph-based clustering algorithm that uses a density estimation technique to identify clusters in the graph. Label Propagation is a simple label propagation algorithm that assigns labels to nodes based on the labels of their neighbors. Async LPA is an asynchronous version of Label Propagation that updates nodes in a random order. Recursive KNN is a label propagation algorithm that uses K-Nearest Neighbor search to propagate labels recursively. Voronoi Clustering is a technique that partitions a point cloud into regions based on the Voronoi diagram of the point cloud. KNNG is the algorithm presented in this study, which uses a proximity-based approach to propagate labels within the graph.

The evaluation of the different label propagation techniques was done by comparing their performance on the test set of synthetic tree point clouds generated using Speedtree. The goal of the evaluation was to determine which technique was the most effective in propagating labels within clusters and improving the labeling of the unlabeled points in the point cloud.

To evaluate the performance of each technique, the F-score for each class label was calculated and compared between the methods. The F-score is a commonly used metric that measures the harmonic mean between precision and recall, where precision is the proportion of true positives over the total predicted positives, and recall is the proportion of true positives over the total actual positives. The F-score is useful in measuring the overall performance of a technique across all class labels, as it takes into account both false positives and false negatives.

It was important to evaluate the techniques for their ability to label all points in the point cloud. These techniques are capable of producing clusters with no partially labeled points contained within the cluster. As a result, all unlabeled points within that cluster remained unlabeled and were removed from the calculations. To account for this, the label coverage of each technique was evaluated.

Another important factor in evaluating the techniques was to compare their performance to the original PointVoxel Transformer segmentation model. The PointVoxel Transformer was trained on a subsample of the point cloud and tested on both a subsample and the entire point cloud. The performance of the label propagation techniques was compared to the performance of the PointVoxel Transformer when the entire point cloud was input to the model. This allowed for a comparison of how well each technique improved upon the labeling of the unlabeled points in the point cloud.

To determine which technique was the most effective, the F-scores, label coverage, and time expenditure of each technique were compared.

IV. Results and Discussion

In this study, we proposed the use of K-Neighbor-Nurtured-Garden (KNNG) for label propagation in synthetic tree point clouds. Table I presents the performance comparison in terms of F-scores for the three classes (trunk, branch, and leaf) and computation time for each method. While the method showed promise in terms of computational efficiency and label coverage, its performance in terms of F-scores for the three classes (trunk, branch, and leaf) was found to be lackluster when compared to other methods. Particularly, the performance drop in trunk points was more pronounced compared to branch and leaf points. In this section, we discuss the potential reasons for this unexpected outcome and provide insights into the limitations of the KNNG method.

One possible reason for the observed performance drop in trunk points could be the inherent nature of the KNNG algorithm, which relies on the k-nearest neighbors to propagate labels. Trunk points, by their nature, tend to be spatially more concentrated and localized in a tree point cloud. As a result, the k-nearest neighbors of a trunk point are more likely to be other trunk points. In the presence of noise or errors in the initial labeling, this local concentration can lead to a reinforcing feedback loop, where incorrect labels propagate and amplify within the trunk points, ultimately affecting the F-score.

Furthermore, the presence of varying point densities within the point cloud may also have affected the performance of the KNNG method. These factors can introduce inconsistencies in the spatial distribution of the points, making it challenging for the algorithm to propagate labels accurately based on the neighborhood information. In particular, the trunk region may be more susceptible to such inconsistencies.

In light of these observations, it appears that the KNNG method may have certain limitations when applied to synthetic tree point clouds, particularly in the context of trunk point

TABLE I
SUMMARY OF STUDY RESULTS: AVERAGE PERFORMANCE METRICS ACROSS 100 TEST POINTCLOUD TREES.

K-NN connectivity inference					
Community Detection Method	Average Additional Inference time (s)	Label coverage	Trunk F-Score	Branch F-Score	Leaf F-Score
FUMO	123.3($\pm$80.9)	99.99%	0.249	0.289	0.743
Spectral Lanczos	4573.9($\pm$1812.2)	60%	0.598	0.72	0.817
AFC	15.1($\pm$6.9)	61.60%	0.746	0.828	0.92
Label Propagation	3.4($\pm$1.4)	71%	0.752	0.832	0.931
Asyn LPA	8.90($\pm$3.53)	62%	0.761	0.832	0.926
Alpha Shapes Surface Reconstruction connectivity inference					
Community Detection Method	Average Additional Inference time (s)	Label coverage	Trunk F-Score	Branch F-Score	Leaf F-Score
FUMO	395.35($\pm$269.34)	99.99%	0.214	0.427	0.586
Spectral Lanczos	20944($\pm$11337.77)	82%	0.402	0.64	0.77
AFC	3.5($\pm$1.3)	97%	0.355	0.624	0.743
Label Propagation	1.7($\pm$0.53)	89%	0.307	0.645	0.75
Asyn LPA	2.1($\pm$0.64)	89%	0.363	0.651	0.765
No Graph Structure					
Method	Average Additional Inference time (s)	Label coverage	Trunk F-Score	Branch F-Score	Leaf F-Score
PVT(2048)	-	2048 pts	0.945	0.859	0.919
PVT(all)	-	100%	0.714	0.783	0.879
rKNN	10.1($\pm$5.6)	100%	0.771	0.801	0.911
Voronoi Clustering	5.34($\pm$1.72)	100%	0.704	0.728	0.8726
KNNG (ours) K=4	3.91($\pm$1.36)	100%	0.525	0.709	0.857
KNNG (ours) K=3	4.75($\pm$1.76)	100%	0.521	0.705	0.853
KNNG (ours) K=2	7.06($\pm$2.93)	100%	0.53	0.713	0.858
KNNG (ours) K=1	7.38($\pm$3.05)	100%	0.569	0.704	0.835

labeling. While the method offers advantages in terms of computational speed and label coverage, its performance in terms of F-scores is found to be inferior to other methods, such as Asyn LPA and Label Propagation. Future research could focus on refining the KNNG algorithm or exploring alternative methods that can better capture the unique spatial characteristics of tree point clouds and achieve more accurate label propagation across all classes.

While the K-Neighbor-Nurtured-Garden (KNNG) method did not perform as expected, other methods utilizing the K-NN connectivity inference such as Asynchronous Label Propagation Algorithm (Asyn LPA) and Label Propagation demonstrated superior performance in terms of F-scores across trunk, branch, and leaf classes.

One of the factors that may have contributed to the superior performance of Asyn LPA and Label Propagation is their ability to better adapt to the complex spatial structures present in tree point clouds. These methods are based on propagating labels through an iterative process, which allows them to effectively capture the underlying structure of the data and propagate labels more accurately. This iterative approach can help mitigate the impact of noise or errors in the initial labeling, as the algorithms continuously refine the labels based on the information from neighboring points.

In addition, the Asyn LPA method allows for asynchronous updating of labels during the propagation process, which can be beneficial in situations where the point cloud exhibits varying point densities, occlusions, or overlapping regions. This asynchronous updating enables the method to be more robust against local inconsistencies in the spatial distribution of points, as it does not rely solely on the immediate neighborhood information for label updates. This flexibility may have contributed to the improved performance of Asyn LPA in comparison to other methods.

Label Propagation also performed well in our study, potentially due to its ability to adapt to the specific characteristics of tree point clouds. Like Asyn LPA, Label Propagation is an iterative method that refines labels based on the information from neighboring points. However, Label Propagation operates in a synchronous manner, updating all labels simultaneously at each iteration. This approach can lead to faster convergence of the algorithm, which might be advantageous in cases where the point cloud exhibits a clear and well-defined structure.

Moreover, both Asyn LPA and Label Propagation methods can be further fine-tuned by adjusting their respective parameters, such as the number of iterations or the influence of neighborhood size. This flexibility allows for the optimization of the methods for the specific characteristics of synthetic tree point clouds, potentially leading to improved performance.

In summary, the superior performance of Asyn LPA and Label Propagation methods in our study can be attributed to their ability to better adapt to the complex spatial structures present in tree point clouds, as well as their iterative nature, which allows for more accurate label propagation. These methods demonstrate the potential to effectively address the challenges associated with labeling synthetic tree point clouds and can serve as a foundation for future research in this area.

Our study revealed that community detection clustering techniques performed significantly better when applied to graphs built using K-NN connectivity inference compared to those constructed using the Alpha Shapes Surface Reconstruction (ASSR) algorithm.

The main difference between the K-NN connectivity inference and ASSR lies in the way they construct the graph structure. The K-NN approach directly connects points based on their proximity in Euclidean space, which makes it more sensitive to local point densities. As a result, K-NN connectivity inference tends to create better-defined clusters that capture the underlying structure of the point cloud more accurately.

On the other hand, ASSR constructs the graph by approximating the underlying surface of the 3D object, which might not always capture the local point densities effectively. This can lead to situations where the reconstructed mesh inaccurately reflects the spatial distribution of points, resulting in less cohesive clusters. Consequently, community detection methods applied to ASSR-based graphs might struggle to propagate labels efficiently and accurately, leading to lower F-scores.

Another possible explanation for the performance difference is that K-NN connectivity inference is more flexible in terms of adapting to various point densities and noise levels. This allows the connectivity to be tailored to the specific characteristics of the point cloud data, whereas ASSR relies on a fixed set of parameters that might not be suitable for all cases.

It is also worth noting that the choice of community detection method can play a significant role in the overall performance of label propagation. Our results showed that some methods, such as Asynchronous Label Propagation Algorithm (Asyn LPA), achieved higher F-scores than others, regardless of the graph construction technique. This suggests that the effectiveness of label propagation is not solely determined by the graph connectivity but also depends on the choice of community detection algorithm.

In conclusion, our findings highlight the importance of selecting an appropriate graph connectivity inference method when using community detection techniques for label propagation in point cloud data. The K-NN connectivity inference approach appears to be better suited for this task, providing more accurate and efficient label propagation compared to the Alpha Shapes Surface Reconstruction algorithm.

V. Conclusion

In conclusion, this study sought to evaluate the performance of various label propagation techniques for improving the labeling of synthetic tree point clouds generated using Speedtree and semantically segmented using a Point-Voxel Transformer model. The goal was to determine the most effective technique for propagating labels within clusters and enhancing the labeling of unlabeled points in the point cloud. We compared the performance of different community detection methods, including modularity-based clustering [27], spectral clustering [28], and the Asynchronous Label Propagation Algorithm (Asyn LPA) [29], applied to graphs built using K-NN connectivity inference and Alpha Shapes Surface Reconstruction (ASSR) algorithm [23], as well as several unstructured-pointcloud based methods. Our evaluation considered factors such as F-scores for each class label, label coverage, and time expenditure.

Our results, based on experiments with 100 test-set synthetic tree pointclouds, demonstrated that community detection clustering techniques applied to graphs built using K-NN connectivity inference consistently outperformed those applied to graphs constructed using the ASSR algorithm. This suggests that the choice of graph connectivity inference method is critical for achieving accurate and efficient label propagation in point cloud data. The K-NN approach, with its sensitivity to local point densities and adaptability to various point densities and noise levels, appears to be better suited for this task compared to the ASSR algorithm.

Furthermore, the choice of community detection method played a significant role in the overall performance of label propagation. Some methods, such as the Asynchronous Label Propagation Algorithm (Asyn LPA), consistently achieved higher F-scores, regardless of the graph construction technique, highlighting the importance of selecting the appropriate community detection algorithm.

Interestingly, the proposed K-Neighbor-Nurtured-Garden (KNNG) method, which was initially expected to perform well, yielded relatively low F-scores (average F-score: 0.76) compared to the Asyn LPA (average F-score: 0.89) and Label Propagation (average F-score: 0.87) methods despite its fast inference time and 100% label coverage. This finding presents an opportunity for future research to further investigate the reasons behind KNNG's suboptimal performance and explore potential improvements to the method, such as incorporating weighting schemes based on point density, or adapting the neighborhood size according to the spatial distribution of points.

In light of our findings, we recommend that future work in point cloud labeling focus on leveraging K-NN connectivity inference and carefully selecting the most suitable community detection method for a given dataset. Additionally, continued exploration and optimization of label propagation techniques, including our proposed KNNG method, could lead to even more accurate and efficient point cloud labeling solutions in the future.

In summary, this study contributes to our understanding of label propagation techniques for point cloud data and offers valuable insights to guide the selection of graph connectivity inference methods and community detection algorithms. The findings provide a foundation for further research and development in the field of point cloud labeling, ultimately contributing to advancements in 3D object recognition, robotics, and computer vision applications.

References

[1] M. C. Hansen, P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland *et al.*, "High-resolution global maps of 21st-century forest cover change," *science*, vol. 342, no. 6160, pp. 850–853, 2013.

[2] G. P. Asner and J. Mascaro, "Mapping tropical forest carbon: Calibrating plot estimates to a simple lidar metric," *Remote Sensing of Environment*, vol. 140, pp. 614–624, 2014.

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[4] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

[5] Y. Xu, T. Fan, M. Xu, L. Zeng, S. Qiao, and Y. Wei, "Spidercnn: Deep learning on point sets with parameterized convolutional filters," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 87–102.

[6] H. Thomas, C. R. Qi, J. E. Deschaud, and B. Marcotegui, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6411–6420.

[7] S. Yang, Y. Li, Z. Zhou, W. He, and J. Deng, "Learning object-centric spatial relations for 3d object detection and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7581–7590.

[8] Y. Liu, P. Jiang, and B. Zhang, "Pvt: Point-voxel transformer for efficient 3d deep learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[9] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, 2018.

[10] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," in *Proceedings of the European Conference on Computer Vision*. Springer, 2018, pp. 268–283.

[11] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*, vol. 96, 1996, pp. 226–231.

[12] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," in *International Journal of Computer Vision*, vol. 59, no. 2. Springer, 2004, pp. 167–181.

[13] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[14] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[15] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[16] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, p. 036104, 2006.

[17] V. A. Traag, "Louvain algorithm: review," *arXiv preprint arXiv:1810.08473*, 2019.

[18] V. A. Traag and J. Bruggeman, "Narrow scope for hubness phenomenon," *Scientific reports*, vol. 1, no. 1, pp. 1–5, 2011.

[19] B. Tesson, T. Reme, and A. Desmoulière, "Network analyses: basic concepts and applications in oncology," *Bulletin du cancer*, vol. 97, no. 10, pp. 1285–1295, 2010.

[20] R. F. Betzel and D. S. Bassett, "The modular organization of human anatomical brain networks: Accounting for the cost of wiring," *Network neuroscience*, vol. 2, no. 3, pp. 222–241, 2018.

[21] G. Cordasco and L. Gargano, "Community detection via semi-synchronous label propagation algorithms," in *Business Applications of Social Network Analysis (BASNA), 2010 IEEE International Workshop on*. IEEE, 2010, pp. 1–8.

[22] F. Parés, D. Garcia-Gasulla, J. Borge-Holthoefer, and Y. Moreno, "Fluid communities: A competitive and highly scalable community detection algorithm," *arXiv preprint arXiv:1703.09307*, 2017.

[23] H. Edelsbrunner and E. P. Mücke, "Shape reconstruction with delaunay triangulation," *IEEE Transactions on Software Engineering*, vol. SE-9, no. 11, pp. 128–141, 1983.

[24] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Symposium on Geometry processing*, vol. 7. Citeseer, 2006, pp. 61–70.

[25] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE transactions on visualization and computer graphics*, vol. 5, no. 4, pp. 349–359, 1999.

[26] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM siggraph computer graphics*, vol. 21, no. 4. ACM, 1987, pp. 163–169.

[27] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[28] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[29] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," in *Physical review E*, vol. 76, no. 3. APS, 2007, p. 036106.