

Neural Hierarchical Decomposition for Single Image Plant Modeling

Zhihao Liu^{1,2} Zhanglin Cheng³ Naoto Yokoya^{1,2,*}

¹ The University of Tokyo ² RIKEN AIP ³ SIAT, Chinese Academy of Sciences

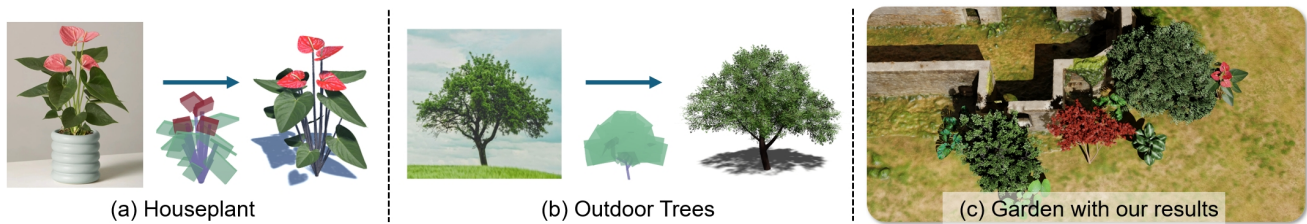


Figure 1. We propose a novel learning-based framework that automatically generates high-quality 3D plant models from single-view images. Our method sequentially combines hierarchical structure learning with parametric modeling based on botanical priors, producing usable, structured plant assets ready for immediate use in practical applications. Through learning the decomposition of box hierarchies at different levels of detail (LoD), our method offers a comprehensive solution that can adapt to two prominent categories of plants: (a) **Houseplants** and (b) **larger trees**. (c) shows a garden with several of our results directly assembled.

Abstract

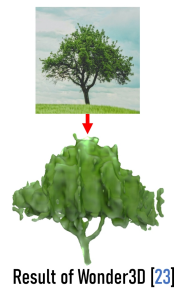
Obtaining high-quality, practically usable 3D models of biological plants remains a significant challenge in computer vision and graphics. In this paper, we present a novel method for generating realistic 3D plant models from single-view photographs. Our approach employs a neural decomposition technique to learn a lightweight hierarchical box representation from the image, effectively capturing the structures and botanical features of plants. Then, this representation can be subsequently refined through a shape-guided parametric modeling module to produce complete 3D plant models. By combining hierarchical learning and parametric modeling, our method generates structured 3D plant assets with fine geometric details. Notably, through learning the decomposition in different levels of detail, our method can adapt to two distinct plant categories: outdoor trees and houseplants, each with unique appearance features. Within the scope of plant modeling, our method is the first comprehensive solution capable of reconstructing both plant categories from single-view images.

1. Introduction

Vegetation is an essential part of natural scenes. Their high-quality 3D models can greatly enhance the realism of various applications such as video games, movies, and land-

scape design. Traditional approaches have relied on procedural modeling [12, 31, 33–35, 46] to generate random 3D plant models from scratch and demonstrated considerable capability, but their complexity often presents a significant barrier to non-expert users, limiting their practical utility. Alternative approaches explore more direct reconstruction methods, such as reconstructing plant models from 3D point clouds [6, 18, 44]. However, they are hindered by the expensive and time-consuming nature of data acquisition. Thus, this limitation has motivated the exploration of single image-based reconstruction as a more flexible and accessible choice.

Compared to typical man-made objects, reconstructing plants solely from single images is considerably more challenging due to their intricate branching structures and significant self-occlusion. In recent years, a number of methods in generative AI have been proposed for single-view 3D reconstruction, including Wonder3D [26] and several concurrent works [15–17]. Although working well on smooth-surfaced objects, these pure AI-driven methods still face a critical limitation in plant modeling: i.e., the inability to capture fine details and high-resolution features. Thus, when applied to 3D plants that have intricate structures, they often produce problematic, noisy results. The wrapped figure on the left shows an example result of Wonder3D [26]. Apparently, generating high-quality, usable 3D plants models with intricate branch and leaf details ex-



Email: liuzh96@outlook.com (Zhihao Liu)

* Corresponding author.

ceeds their capabilities.

To truly obtain production-ready 3D plant assets from single images, recent efforts have attempted to integrate graphics techniques with AI workflows. These specialized algorithms [11, 22] primarily focus on reconstructing outdoor leafy trees and typically employ a two-stage process: neural network-based inference of intermediate 3D envelopes, followed by procedural rule-based geometry detail propagation [31]. However, these methods often lack either full automation or sufficient flexibility to capture diverse tree shapes. Moreover, they fail to address the distinct requirements of houseplants, which demand more precise modeling of individual leaves and twigs.

In this paper, we present a novel generic framework for reconstructing high-quality 3D models for both houseplants and trees from single-view images. The core of our approach lies at a hierarchical box representation that encodes plant structures as a hierarchy of *n-ary* graphs. Each box (i.e., graph node) functions either as a terminal component representing individual plant parts, or as an assembly node that can be recursively decomposed into more detailed sub-structures. These boxes are implemented as oriented bounding boxes of plant parts and are associated with a small set of botanical parameters that capture essential structural characteristics of their enclosed plant components. Our reconstruction pipeline processes a single plant image through a specialized graph decoder, which recursively builds the hierarchical structure from coarse to fine details. The final stage employs parametric geometry modeling to transform this abstract box hierarchy into detailed, high-quality 3D plant models suitable for practical applications.

To handle both houseplants and outdoor trees, we implement different levels of detail (LOD) in our hierarchical decomposition. For **houseplant** reconstruction, terminal nodes directly represent individual elements such as leaves, flowers, or short branch stems (see Fig. 1 (a)). However, for **outdoor trees**, where complete reconstruction of dense foliage would be computationally intractable, we terminate the decomposition of hierarchical box structures early at a reasonably larger scale. As shown in Fig. 1 (b), each terminal box for the foliage region (in green) corresponds to a sub-tree component, and thus the entire tree structure is represented as a small number of bounding boxes in an abstract manner. The final complete 3D tree geometry can be constructed by procedurally growing branches and leaves within each sub-tree box.

In summary, our main contributions include:

- A novel neural decomposition framework that learns compact, hierarchical box representations of plants from single-view images, effectively capturing their structural, morphological, and botanical parameters.
- The first comprehensive solution for high-quality 3D re-

construction of both houseplants and outdoor trees from single photographs, addressing the distinct morphological requirements of each category through an adaptive, hierarchical modeling approach.

2. Related Work

Realistic Plant Reconstruction. Early approaches to obtain geometric plant models primarily focus on procedural generation, where the plant growth is manipulated by structural rules (e.g., L-systems [34, 35] and self-organization [31, 37]), or by user interactions [20, 23, 49]. As sensing devices continue to advance, reconstructing 3D plants from real-world data modalities has become a mainstream of research. Existing works mainly use LiDAR or multi-view images to get dense 3D point clouds, which are then converted into plant models through skeleton connection [6, 24, 39, 44], lobe extraction [25], and deep learning [18]. The most related to our work is single-image tree reconstruction [11, 14, 22, 40]. In contrast to 3D point scans, single-view images are much easier to acquire. However, these methods present several limitations: First, a common issue is that all these methods are only designed for outdoor trees and cannot adapt to houseplants. Second, these methods either require extra user interactions (e.g., drawing) [22, 40], or lack flexibility to handle complex tree shapes [11, 14]. To the best of our knowledge, none of the existing single-view-based approaches can offer a generic framework that can adapt to both outdoor trees and smaller houseplants simultaneously.

Single-Image 3D Reconstruction. With the advancements in deep learning, single-view shape reconstruction has seen rapid growth in recent years. Early approaches mainly focus on low-precision representations like 3D voxels [43, 45]. More recently, with the support of diffusion models, a new wave of techniques has demonstrated the potential of using novel view synthesis to reconstruct 3D mesh models from single images, involving Wonder3D [26] and other concurrent works [15–17, 19]. However, these methods typically rely on optimizing a NeRF representation [28], and thus the resulting geometries are often a watertight surface with inseparable structures. While they perform well on objects with smooth surfaces, they struggle to produce high-quality, usable 3D plant models, particularly for trees which feature extremely complex branching structures and foliage details (Please see Fig. 9).

Deep Learning for Part Assembly. Many objects in the real world can be naturally decomposed into a set of smaller sub-parts. A series of studies tried to leverage the concept of part assembly to decompose the 3D shapes in the form of sequential/hierarchical parts [13, 32, 42, 47, 50] through deep neural networks. Subsequent works further explored their applications in tackling various 3D tasks, such as shape generation [5, 29, 30, 48], shape segmentation [9, 41], and ob-

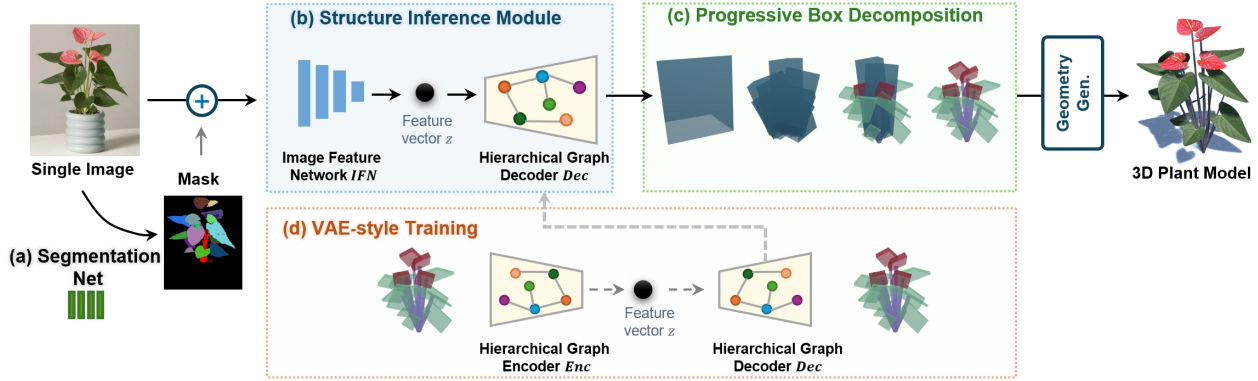


Figure 2. Overview of our framework. (a) Starting from the input photograph, we first use a segmentation module to obtain a mask image. (b) Then, after stacking the photograph and its mask channel-wise, we employ a structure inference module to progressively yield a (c) hierarchical box decomposition from coarse to fine. Finally, the detailed 3D plant model is synthesized automatically based on the last box decomposition by using a parametric modeling method. Our hierarchy inference network consists of two separately trained modules: an image feature network IFN to map the image to a latent feature vector z , while a hierarchical graph decoder Dec tries to decode the box hierarchies from this feature vector. (d) We also use a mirrored graph decoder Enc to achieve the VAE-style training for the Dec , in which the Enc and Dec learn a reversible mapping between the plant hierarchies and the latent feature space.

ject detection [1]. However, existing works mainly target man-made objects with simple structures (e.g., tables and chairs). In this paper, we argue that plants can be abstracted into a certain form of part assembly as well, which can serve as an effective medium to guide the reconstruction process for botanical plants.

3. Overview

In this paper, we present a generic learning-based framework to reconstruct both houseplants and leafy trees from single-view images. Fig. 2 illustrates the overall pipeline of our reconstruction algorithm by taking a houseplant (anthurium) as an example. We introduce hierarchical box representations that serve as a high-level description of the plant structures by decomposing them into a set of boxes. We show that the hierarchical boxes are capable enough of preserving complex structural features of various plant species by using a variable number of boxes.

Our method consists of three steps: First, as data preparation, we employ a **segmentation network** to output a mask image where each pixel is assigned a label. This mask is also used to remove the photograph’s background. Then, the input image and its mask are stacked channel-wise and fed to a **structure inference module** to progressively produce the hierarchical box decomposition from coarse to fine. Fig. 3 (a) also illustrates an example of the box decomposition procedure at different stages. Finally, the detailed 3D plant geometries can be automatically synthesized from the last level of the hierarchical boxes by applying a parametric modeling method based on the parameters and labels of each terminal box.

As shown in Fig. 2(b), our structure inference module

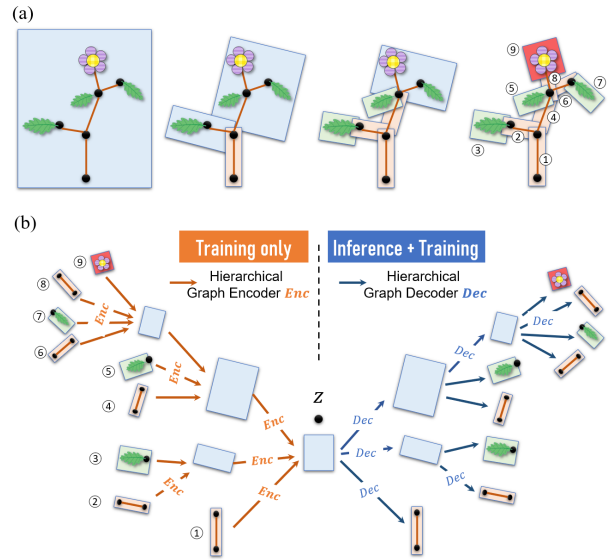


Figure 3. (a) Hierarchical box decomposition of a plant structure at different inference stages. (b) The VAE-style training uses mirrored Enc and Dec to map between plant structures and latent feature space in opposite directions.

further consists of two successive sub-parts: (1) The first part is an *image feature network* IFN that takes the photograph and its mask as input to extract a feature vector z in latent space. (2) Then, a *hierarchical graph decoder* Dec transforms this feature vector z back into hierarchical boxes. As shown in the right part of Fig. 3 (b), the hierarchical graph decoder Dec recursively expands a parent feature vector back into a set of child boxes in a bottom-

up fashion. The decomposition process starts from the root (i.e., the z encoded by IFN) and stops until no intermediate node is decomposed. In our implementation, these two sub-modules are separately trained to obtain better inference ability. As shown in Fig. 2 (d), we use another mirrored hierarchical graph encoder Enc to help train the decoder Dec in a variational autoencoder (VAE) manner, in which the Enc and Dec learn a reversible mapping between the plant hierarchies and the latent feature space (see the left part of Fig. 3 (b)). After training the Enc and Dec , we can subsequently train the image feature network IFN based on the latent feature space learned by Enc .

Additionally, this hierarchical box representation is also applicable to outdoor trees through performing decomposition in different levels of detail, in which terminal boxes are instanced as a sub-tree instead of a single leaf (Sec. 4.3). Please refer to our supplementary video for the animated demonstration of the decomposition procedure and resulting 3D models.

4. Methodology

4.1. Definition

Hierarchical Box Representation for Plants. We introduce a shape representation called Hierarchical Boxes to describe the plant shapes in a light-weight and abstract manner. As shown in Fig. 3 (a), we assume that a plant can be decomposed into a hierarchical assembly of boxes that encodes both the geometry and the structure of a plant. This box hierarchy is represented as an n -ary tree, where each node is a single plant component or component assembly. Starting from the largest box that corresponds to the entire plant structure, this hierarchical decomposition works recursively in a bottom-up fashion until reaching the most fine-grained plant components at leaf nodes.

In our implementation, a box hierarchy can be written as $\mathbf{H} := (\mathbf{B}, \mathbf{E})$, composed of a set of boxes $\mathbf{B} := \{B_1, B_2, \dots\}$ that describes the geometry of a plant part, and a set of edges \mathbf{E} that describes the adjacency relationships between the plant parts. Each box B_i corresponds to a sub-part of the plant, and can be characterized by an oriented bounding box: $B_i := (c, r, s, l, p)$, where c is the world coordinate of the box center, r is the rotation of the box defined as a quaternion, s is the length of the box along each axis, l is a semantic label, and p is a vector of botanical parameters.

Specifically, label l indicates the semantics of the box, which can be either an intermediate node (i.e., an expandable node that can be further expanded into a graph at the next lower level), or other terminal node types (for houseplants, a terminal node can be a branch, leaf or flower; while for outdoor trees, it can be a sub-tree part or a main trunk branch). In addition, we also use a vector p to store

the botanical parameters for the box, which describes the appearance characteristics of the corresponding plant parts (such as the bending angles of leaf surfaces), and can be used to construct the detailed 3D geometries by using our 3D modeling algorithm. Note that p is only meaningful when the box is a terminal node, otherwise, it is set to all zeros. We set $|p| = 12$ which means each box can store up to 12-dimensional additional parameters. Please refer to **supplementary material** for a detailed list of the botanical parameters p .

4.2. Neural Decomposition of Plant Parts

In the following, we will introduce the details of network architectures and training setups.

4.2.1. Segmentation

The first step is to generate segmentation masks from real-world plant photographs. Considering the different characteristics of houseplants and outdoor trees, we apply instance segmentation and semantic segmentation for them, respectively. In our workflow, we employ UPerNet with Swin-Transformer [21] to produce the mask images, which capably serves as a general-purpose backbone for both instance and semantic segmentation. The advantage of using segmentation masks is to reduce the domain gap between real-world photographs and our synthetic ones by directly providing semantic information of plant structures, and also serve as a binary mask to remove the background of the input photographs before stepping into the next inference stage.

4.2.2. Image Feature Network

After segmentation, we then stack the input RGB photograph (with the background removed) and its segmentation mask in a channel-wise manner, and employ an image feature network IFN to map it to a latent feature vector z . The image encoder is based on a ResNet-50 architecture [7], which has been demonstrated as an effective module to extract features from 2D images.

4.2.3. Hierarchical Graph Decoder

The core component to achieve neural box decomposition is a hierarchical graph decoder Dec which recursively expands nodes from the root until the full plant structure is reconstructed. It consists of two independent sub-decoders: Dec_g and Dec_{box} .

(1) The first **sub-decoder** Dec_g adopts a common graph generative model (MolGAN [2]) as the network backbone, which is able to ingest a latent vector and output a graph structure. As shown in Fig. 4(a), our decoder Dec_g decodes the feature vector z_i of a parent box B_i into a set of child boxes $\{B_k\}$:

$$Dec_g(z_i) = \{(z_k, l_k, e_k) | B_k \in C_i\}, \quad (1)$$

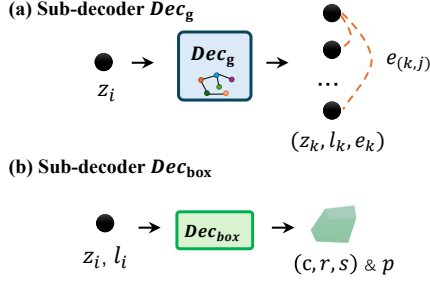


Figure 4. The hierarchical graph decoder Dec contains two sub-decoders: (a) Dec_g is to expand the feature vector of a node into a sub-graph, where each graph node also corresponds to a new feature vector z_k . (b) Dec_{box} is to translate the implicit feature vector of a node into its actual spatial box shape.

where C_i are the child parts. Each child box $B_k \in C_i$ is represented by a feature vector z_k , semantic label l_k and existence probability e_k . Moreover, the Dec_g also predicts the edge's existence probability $e_{(k,j)}$ between every two child boxes $B_k, B_j \in C_i$.

In our implementation, the Dec_g always decodes a fixed maximum number of $n = 12$ child boxes and n^2 edges. Boxes and edges with their existence probability $e_k < 0.5$ or $e_{(k,j)} < 0.5$ are discarded. The dimensionality of any feature vector z_i is set to 256. The decomposition process starts from the root node whose feature vector is denoted as z_0 , and continues until there are no remaining intermediate boxes.

(2) Additionally, to transform the implicit feature vector of the node into the explicit description of target box shape, we use a second **sub-decoder** Dec_{box} which is a multiple-layer perceptron (MLP). As shown in Fig. 4(b), Dec_{box} takes as input a feature vector z_i and its label l_i to recover the spatial box shape $B_i = (c, r, s)$ as well as the botanical parameters p of the corresponding plant part. This Dec_{box} module can be used to reconstruct the box shape for both intermediate nodes and terminal nodes during the inference process. However, the predicted botanical parameters p have specific meanings only for terminal nodes.

4.2.4. Training

We train the two sub-modules of the structure inference module (i.e., the IFN and Dec) in a separate manner.

Training of Image Feature Network. The first module IFN tries to map a photograph I to a ground-truth feature vector z , thus it can directly be trained via the following loss:

$$L_{\text{img}}(I) = \|z - IFN(I)\|^2. \quad (2)$$

Note that the IFN should be trained after the hierarchical graph decoder and encoder (Dec and Enc) since its training relies on the latent feature space produced by the Enc .

VAE-style training for hierarchical graph decoder. The

Dec is trained in a complicated way. Briefly, we followed a variational autoencoder (VAE) fashion introduced by StructureNet [29] to train this network. As shown in the left part of Fig 3(b), we additionally used a hierarchical graph encoder Enc to map a plant structure S to a latent feature vector z . This graph encoder is simply implemented as a mirrored architecture of our hierarchical graph decoder Dec . Therefore, we aim to train the encoder and decoder to perform a reversible mapping between a plant structure S and a feature vector z . To train this graph VAE, the loss can be written as follows:

$$L_{\text{final}} = \alpha L_{\text{vae}}(S) + \beta L_{\text{recon}}(S) + \gamma L_{\text{sem}}(S) + \lambda L_{\text{exist}}(S), \quad (3)$$

where L_{vae} is the conventional variational regularization loss based on KL divergence, which encourages the structure to be embedded in a smooth latent space. This loss term has been explained by [10] at great length. L_{recon} , L_{sem} and L_{exist} denote the reconstruction loss, semantic loss, and existence loss, respectively. We then define the last three loss terms in detail.

(1) **Reconstruction Loss.** The L_{recon} measures the geometric similarity between the hierarchical bounding boxes of the ground-truth plant structure S and the reconstructed plant $S' = Dec(Enc(S))$:

$$L_{\text{recon}}(S) = \sum_{(B_i, B'_i) \in (S, S')} \text{chf}(\tau(B_i), \tau(B'_i)), \quad (4)$$

where $\text{chf}(\cdot, \cdot)$ denotes the Chamfer distance [3] that can effectively evaluate the geometric distance between two shapes. B_i and B'_i are the boxes in plant structures S and S' , respectively. $\tau(B_i) = T(B_i)N$, where N is a fixed pre-sampled set of points on a unit box's surface, and $T(B_i)$ is the transformation matrix that transforms a unit box to the bounding box B_i of a plant part.

(2) **Semantic Loss.** The L_{sem} evaluates the consistency of both semantic labels l and shape parameters p between the ground truths and reconstructions:

$$L_{\text{sem}}(S) = \sum_{(B_i, B'_i) \in (S, S')} \left[\sigma(l_i, l'_i) + \|p_i - p'_i\|^2 \right], \quad (5)$$

where $\sigma(\cdot, \cdot)$ is the cross entropy loss function. l_i and l'_i are the ground truth and reconstructed semantic labels of the matched boxes B_i and B'_i , while p_i and p'_i are the shape parameters describing our generative geometric meshes.

(3) **Existence loss.** L_{exist} consists of two terms, and uses a cross-entropy to ensure the existence of all the *boxes* and *edges* is accurately reconstructed. It specifically encourages each box B'_j or each edge $E'_{(j,k)}$ in the reconstructed plant structure S' can find a match in the ground truth plant S :

$$L_{\text{exist}} = \sum_{B'_j \in S'} \sigma(e'_j, I(B'_j, S)) + \sum_{E'_{(j,k)} \in S'} \sigma(e'_{(j,k)}, I(E'_{(j,k)}, S)) \quad (6)$$

where e'_j and $e'_{(j,k)}$ are the predicted existence probabilities of a box B'_j and an edge $E'_{(j,k)}$, respectively, which are described in Chap. 4.2.3. $I(\cdot, \cdot)$ is an indicator function that returns 1 if a reconstructed box or an edge can find a match in the ground truth plant S .

4.3. Plant Geometry Construction

Once decomposition is completed, our final stage is to construct the complete 3D plant models. Conditioned on the hierarchical boxes inferred by AI modules, we leverage two parametric modeling strategies to synthesize geometric details for each plant category at different levels of detail (LoD).

Outdoor Trees. Trees often feature heavy foliage, making it unfeasible to conduct leaf-level recognition for a distant single image [11]. Thus, as shown in Fig. 5(a), the decomposition of hierarchical boxes terminates early at a larger size to capture the key structural features, resulting in several sub-tree boxes for foliage (in green), and a few trunk branches (in purple).

Fig. 5 also illustrates the steps for the progressive generation of tree geometries from hierarchical boxes. Our approach is inspired by a point cloud-based reconstruction method [25]. Starting from the lowest position, we first traverse all purple branch boxes using the minimal spanning tree (MST) algorithm, forming the trunk skeleton in a bottom-up manner (Fig. 5 (b)). Then, based on the trunks, we further propagate foliage details (i.e., twigs and leaves) strictly within the green boxes by applying a parametric L-system module [27, 38] (Fig. 5 (c)). The L-system used here is a popular method that can effectively simulate the developmental growth of branches for various tree species based on predefined botanical rules. During growth, the branches reaching outside the boxes are directly discarded, ensuring the tree shapes align well with the input volumes. In addition to limiting the growth space, each box is also associated with several botanical parameters p predicted by Dec_{box} , which depict various growth features of this area (e.g., twig density, branch angles, leaf size) and will be used by L-system to produce corresponding sub-trees. Please refer to **supplementary materials** for the full list of botanical parameters used.

Houseplants. The proposed hierarchical boxes can also adapt to depicting houseplants. As shown in Fig. 6 (a), we construct their models at a finer-grained component level.

Specifically, we represent individual leaves or petals based on parametric deformable surfaces [36], which leverage extensive botanical priors to biologically define templated leaf shapes for various species. The leaf shapes can be flexibly morphed based on the predicted botanical parameters p of each box, enabling us to create leaf surfaces close to the input images. Fig. 6 (b) shows two examples of morphing a simple leaf by changing their radial/tangential

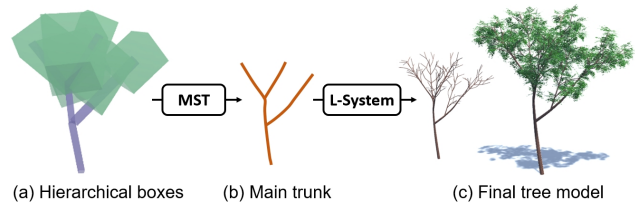


Figure 5. **Tree geometry generation.** (a) Box structure of a tree. (b) Extracted main trunk. (c) The final 3D tree model with new branches propagating within the box space.

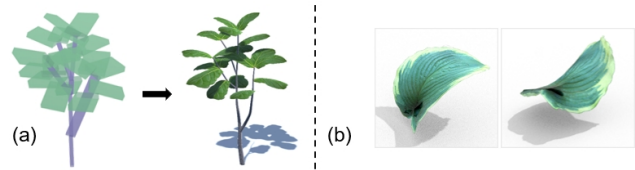


Figure 6. **Houseplant geometry generation.** (a) We construct the houseplant model at a finer-grained component level. (b) The leaves are implemented as parametric deformable surfaces that can be controlled by the botanical parameters predicted by sub-decoder Dec_{box} .

bending angles. Additionally, the size, orientation, and position of each plant component are directly inherited from the corresponding box geometry. The entire plant geometry is also progressively constructed starting from the root using MST, ensuring the neighboring components are accurately connected and preventing the formation of cycles.

5. Experimental Evaluation

We conducted all experiments on a computer equipped with an Intel Core i9-12900H processor with 3.2GHz, 16GB RAM. The network was trained offline on an NVIDIA RTX 3080Ti GPU with 12GB memory. We implemented our hierarchical graph network in PyTorch, and developed the parametric modeling algorithms for 3D plants in Unity.

5.1. Results

Dataset. Due to the difficulty of collecting real-world 3D plants, generating synthetic plant dataset is widely adopted in many relevant tasks, such as point cloud reconstruction [18] and foliage detection [4]. Based on this idea, we implemented two powerful algorithms (self-organization [31] and L-system [38]) for automatically synthesizing diverse 3D plant models of different species.

The final dataset contains 12 tree species (e.g., Maple, Oak, Pine), and 9 common houseplant species (e.g., Anthurium, Monstera Deliciosa), respectively. We trained each species separately. There are 2k unique plants per species in the training dataset, and 1k for testing. Please refer to the **supplementary material** for more details on the dataset.

Table 1. **Quantitative evaluation of network performance using two metrics:** Chamfer distance [3] and Hausdorff distance[8]. We use them to measure the similarities of the **box structures** between the ground-truths and our reconstructions. We computed the root mean square deviation (RMSE) and the standard deviation (SD) of both metrics on the test set. For convenience, the heights of all the plants are normalized to 1.0.

Target	Chamfer Dist ↓	Hausdorff Dist ↓
Houseplant	0.065 (± 0.013)	0.092 (± 0.034)
Outdoor Trees	0.082 (± 0.025)	0.136 (± 0.069)
Average	0.073 (± 0.019)	0.113 (± 0.051)

Network performance. We first evaluate the reconstruction performance of our proposed network on the test set. In Table. 1, we quantitatively evaluate the reconstruction quality of box structures using two metrics: Hausdorff distance [8] and Chamfer distance [3]. The two distances are computed by treating the vertices of box structures as a sparse 3D point cloud. The results demonstrate that our network can effectively decode the proper box structures from the latent space.

Reconstruction results of real plants. To observe the ability of our method to generate 3D plants from real-world photographs, Fig. 7 and Fig. 8 present some results for trees and houseplants, respectively. The input images are either collected from public resources or captured by a smartphone camera. The results demonstrate the effectiveness of our method in decomposing the underlying hierarchy for different plant species, and its robust ability to yield realistic 3D plant models with high geometry details. Additional results can be found in the **supplementary material**.

5.2. Comparison

In this section, we focus on qualitative comparisons with two types of recent methods. For quantitative comparisons and more detailed analysis, please also refer to the **supplementary material**.

(1) Diffusion-based single image to 3D. With advances in diffusion models, many approaches were proposed in the CV community for reconstructing 3D models from single images, especially in the past two years [15, 16, 19, 26]. Existing works typically use diffusion models to generate novel views from single images, and then optimize them into NeRF-like 3D geometries.

Fig. 9 visually compares our method with two recent representative works [16, 26]. These methods produced problematic results when applied to 3D plants. In particular, for outdoor trees which typically have complex branching structures and internal occlusions, their results are often noisy volumes lacking fine details (see Fig. 9 (c-d)). Moreover, the outputs of these methods are structurally inseparable,

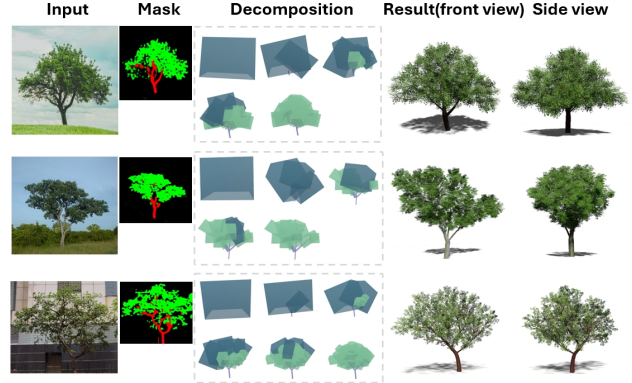


Figure 7. **Generating outdoor trees from real photographs.** From left to right: input image, segmentation, hierarchical box decomposition, and final 3D reconstruction under front and side views.

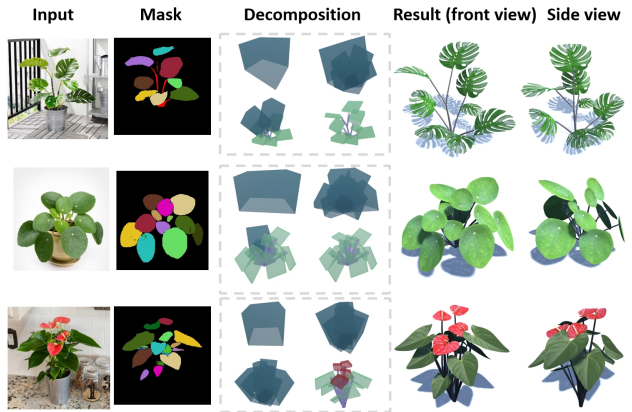


Figure 8. **Generating houseplants from real photographs.** From left to right: input image, segmentation, hierarchical boxes at different decomposition levels, and our reconstructed plant models observed from different views.

arable, which further limits their practical use or further editing. In contrast, our approach overcomes these challenges by sequentially combining the hierarchical learning and parametric modeling to produce high-quality 3D plant assets from coarse to fine. The fine details of our results also allow them to be immediately applicable in 3D applications, such as games.

(2) Plant reconstruction methods. Reconstructing plants from single images is ill-posed due to their high structural complexity. Thus, early approaches [22, 40] are often semi-automatic, requiring users to perform extra user interaction (e.g., drawing annotations) to guide 3D inference. More recently, Li et al. [11], through learning an intermediate representation called Radial Bounding Volume (RBV), presents a fully automatic approach for single image plant reconstruction.

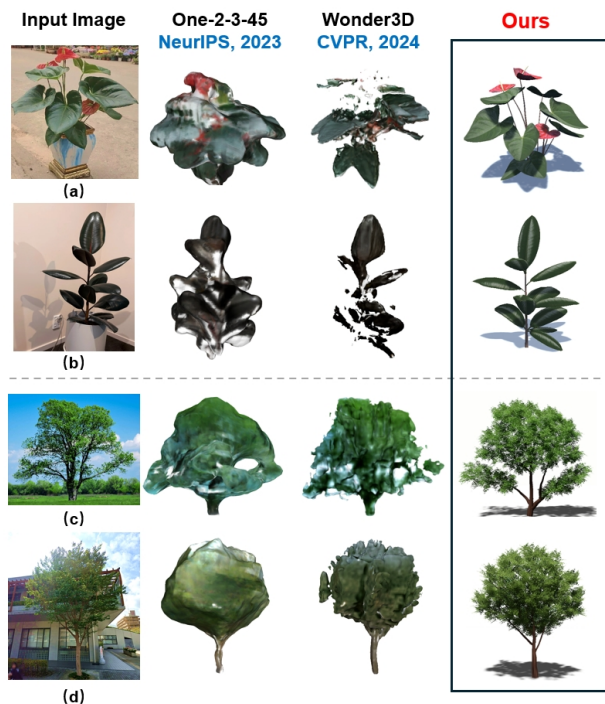


Figure 9. **Comparison with recent single-view 3D reconstruction methods:** One-2-3-45 [16] and Wonder3D [26]. These methods work well for smooth-surfaced objects, but when applied to 3D plants that have intricate structures, they often fail to produce usable 3D models. Especially for trees (c-d), these methods tend to generate problematic meshes that lack structural details. In contrast, our approach is able to produce high-quality 3D plant assets that are ready for instant use in practical applications.

Fig. 10 compares this RBV-based method [11] with our approach. Their method extracts the RBV from input image by MLPs, and then uses it to guide the tree growth. Note that since the method [11] is not open-sourced, we adopted their RBV structure and trained on a more advanced ResNet network than the one used in their original paper. However, as shown in Fig. 10(a), this RBV representation lacks sufficient flexibility to describe challenging tree shapes (such as the holes and long branches within the red box). Additionally, existing methods are only designed for trees, thus unable to adapt to other plant categories - houseplants (see Fig. 10(b)). In contrast, our approach offers higher expressiveness in depicting tree structures, and provides a versatile solution to handle both trees and houseplants.

6. Conclusion

In this paper, we introduced a novel method for generating high-quality 3D plant assets from single-view images. We leverage a hierarchical box representation to effectively capture the 3D structures of plants and serve as a light-

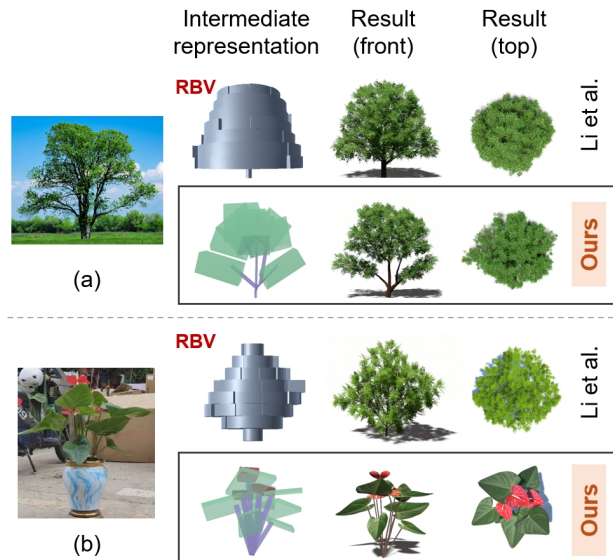


Figure 10. **Comparison with a recent specialized work on 3D tree reconstruction [11]**, which infers a Radial Bounding Volume (RBV) from a single image to guide the tree growth. However, the RBV representation lacks sufficient flexibility to describe diverse tree shapes, such as (a) trees with holes and long branches. In addition, their approach cannot adapt to (b) houseplants. For a rough comparison, we still use this method to generate a result for this houseplant. Our method presents a more expressive hierarchical box representation and can handle both plant categories.

weight medium to guide the subsequent geometry synthesis. By integrating hierarchical structure learning with parametric modeling algorithms, our method is able to generating realistic 3D plant models with fine geometric details, enabling direct use or easy editing in external 3D applications. In addition, compared with existing specialized plant modeling methods (e.g., [11]), our method provides the first generic solution for reconstructing two primary plant categories (trees and houseplants) from a single-view image by learning their decomposition at different levels of detail. Experimental results demonstrated the effectiveness of the proposed method in image-driven plant modeling.

In the future, we plan to learn richer botanical information from the input images, and expand the training dataset to encompass more species. Moreover, we believe that this AI-CG synergy workflow has the potential for generating practically usable, structured 3D models with fine details for many other object categories, such as buildings.

Acknowledgments. We thank the anonymous reviewers for their valuable suggestions. This work is supported in part by the following programs: JST, FOREST under Grant Number JPMJFR206S; RIKEN Junior Research Associate (JRA) Program; NSFC (U21A20515); Guangdong Major Project of Basic Research (2023B0303000016).

References

- [1] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 392–401, 2020. 3
- [2] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *ICML workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018. 4
- [3] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 605–613, 2017. 5, 7
- [4] Adnan Firoze, Cameron Wingren, Raymond A Yeh, Bedrich Benes, and Daniel Aliaga. Tree instance segmentation with temporal contour graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6
- [5] Lin Gao, Jia-Mu Sun, Kaichun Mo, Yu-Kun Lai, Leonidas J Guibas, and Jie Yang. Scenehgn: Hierarchical graph networks for 3d indoor scene generation with fine-grained geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023. 2
- [6] Jianwei Guo, Shibiao Xu, Dong-Ming Yan, Zhanglin Cheng, Marc Jaeger, and Xiaopeng Zhang. Realistic procedural plant modeling from multiple view images. *IEEE transactions on visualization and computer graphics (TVCG)*, 26(2): 1372–1384, 2018. 1, 2
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 770–778, 2016. 4
- [8] Pengdi Huang, Liqiang Lin, Fuyou Xue, Kai Xu, Danny Cohen-Or, and Hui Huang. Hausdorff point convolution with geometric priors. *arXiv preprint arXiv:2012.13118*, 2020. 7
- [9] Jeonghyun Kim, Kaichun Mo, Minhyuk Sung, and Woon-tack Woo. Seg&struct: The interplay between part segmentation and structure inference for 3d shape parsing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1226–1235, 2023. 2
- [10] Thomas N Kipf and Max Welling. Variational graph autoencoders. *NIPS Workshop on Bayesian Deep Learning*, 2016. 5
- [11] Bosheng Li, Jacek Kałuzny, Jonathan Klein, Dominik L Michels, Wojtek Pałubicki, Bedrich Benes, and Sören Pirk. Learning to reconstruct botanical trees from single images. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 2021. 2, 6, 7, 8
- [12] Hongjun Li, Jianwei Guo, Oliver Deussen, and Xiaopeng Zhang. Tree growth modelling constrained by growth equations. In *Computer Graphics Forum (CGF)*, 2018. 1
- [13] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, pages 1–14, 2017. 2
- [14] Yuan Li, Zhihao Liu, Bedrich Benes, Xiaopeng Zhang, and Jianwei Guo. Svdtree: Semantic voxel diffusion for single image tree reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4692–4702, 2024. 2
- [15] Yukang Lin, Haonan Han, Chaoqun Gong, Zunnan Xu, Yachao Zhang, and Xiu Li. Consistent123: One image to highly consistent 3d asset using case-aware diffusion priors. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM)*, pages 6715–6724, 2024. 1, 2, 7
- [16] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 7, 8
- [17] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision (CVPR)*, pages 9298–9309, 2023. 1, 2
- [18] Yanchao Liu, Jianwei Guo, Bedrich Benes, Oliver Deussen, Xiaopeng Zhang, and Hui Huang. Treepartnet: neural decomposition of point clouds for 3d tree reconstruction. *ACM Transactions on Graphics (TOG)*, 2021. 1, 2, 6
- [19] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2023. 2, 7
- [20] Zhihao Liu, Ce Shen, Zhi Li, Tingyu Weng, Oliver Deussen, Zhanglin Cheng, and Dangxiao Wang. Interactive modeling of trees using vr devices. In *International Conference on Virtual Reality and Visualization (VR)*, pages 69–75, 2019. 2
- [21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 10012–10022, 2021. 4
- [22] Zhihao Liu, Kai Wu, Jianwei Guo, Yunhai Wang, Oliver Deussen, and Zhanglin Cheng. Single image tree reconstruction via adversarial network. *Graphical Models*, 2021. 2, 7
- [23] Zhihao Liu, Yu Li, Fangyuan Tu, Ruiyuan Zhang, Zhanglin Cheng, and Naoto Yokoya. Deeptreesketch: Neural graph prediction for faithful 3d tree modeling from sketches. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2024. 2
- [24] Yotam Livny, Feilong Yan, Matt Olson, Baoquan Chen, Hao Zhang, and Jihad El-Sana. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Transactions on Graphics (TOG)*, 2010. 2
- [25] Yotam Livny, Soeren Pirk, Zhanglin Cheng, Feilong Yan, Oliver Deussen, Daniel Cohen-Or, and Baoquan Chen. Texture-lobes for tree modelling. *ACM Transactions on Graphics (TOG)*, 2011. 2, 6
- [26] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang,

- Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR, Highlight)*, pages 9970–9980, 2024. 1, 2, 7, 8
- [27] Radomír Měch and Przemyslaw Prusinkiewicz. Visual models of plants interacting with their environment. In *ACM Trans. on Graphics (Proc. SIGGRAPH)*, pages 397–410, 1996. 6
- [28] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020. 2
- [29] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*, 2019. 2, 5
- [30] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J Mitra, and Leonidas J Guibas. Structedit: Learning structural shape variations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 8859–8868, 2020. 2
- [31] Wojciech Palubicki, Kipp Horel, Steven Longay, Adam Runions, Brendan Lane, Radomír Měch, and Przemyslaw Prusinkiewicz. Self-organizing tree models for image synthesis. In *ACM Trans. on Graphics (Proc. SIGGRAPH)*, page 58, 2009. 1, 2, 6
- [32] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1060–1070, 2020. 2
- [33] Sören Pirk, Ondrej Stava, Julian Kratt, Michel Abdul Masih Said, Boris Neubert, Radomír Měch, Bedrich Benes, and Oliver Deussen. Plastic trees: interactive self-adapting botanical tree models. *ACM Transactions on Graphics (TOG)*, 2012. 1
- [34] P Prusinkiewicz, A Lindenmayer, and J Hanan. The algorithmic beauty of plants. *The virtual laboratory (USA)*, 1990. 2
- [35] Przemyslaw Prusinkiewicz, Mark Hammel, Jim Hanan, and Radomir Mech. L-systems: from the theory to visual models of plants. In *Proceedings of the 2nd CSIRO Symposium on Computational Challenges in Life Sciences*, pages 1–32, 1996. 1, 2
- [36] Adam Runions, Martin Fuhrer, Brendan Lane, Pavol Federl, Anne-Gaëlle Rolland-Lagan, and Przemyslaw Prusinkiewicz. Modeling and visualization of leaf venation patterns. In *ACM SIGGRAPH*. 2005. 6
- [37] Adam Runions, Brendan Lane, and Przemyslaw Prusinkiewicz. Modeling trees with a space colonization algorithm. In *Proceedings of the Third Eurographics conference on Natural Phenomena*, pages 63–70, 2007. 2
- [38] Jerry O Talton, Yu Lou, Steve Lesser, Jared Duke, Radomír Měch, and Vladlen Koltun. Metropolis procedural modeling. *ACM Transactions on Graphics (TOG)*, 2011. 6
- [39] Ping Tan, Gang Zeng, Jingdong Wang, Sing Bing Kang, and Long Quan. Image-based tree modeling. In *ACM Trans. on Graphics (Proc. SIGGRAPH)*, pages 87–es. 2007. 2
- [40] Ping Tan, Tian Fang, Jianxiong Xiao, Peng Zhao, and Long Quan. Single image tree modeling. *ACM Transactions on Graphics (TOG)*, 2008. 2, 7
- [41] Arulmolivarman Thieshanthan, Amashi Niwarthana, Pamuditha Somarathne, Tharindu Wickremasinghe, and Ranga Rodrigo. Hpgnn: Using hierarchical graph neural networks for outdoor point cloud processing. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 2700–2706, 2022. 2
- [42] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 829–838, 2020. 2
- [43] Haozhe Xie, Hongxun Yao, Shengping Zhang, Shangchen Zhou, and Wenxiu Sun. Pix2vox++: Multi-scale context-aware 3d object reconstruction from single and multiple images. *International Journal of Computer Vision (IJCV)*, 2020. 2
- [44] Hui Xu, Nathan Gossett, and Baoquan Chen. Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Transactions on Graphics (TOG)*, 2007. 1, 2
- [45] Shuo Yang, Min Xu, Haozhe Xie, Stuart Perry, and Jiahao Xia. Single-view 3d object reconstruction from shape priors in memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3152–3161, 2021. 2
- [46] Lei Yi, Hongjun Li, Jianwei Guo, Oliver Deussen, and Xiaopeng Zhang. Light-guided tree modeling of diverse biomorphs. In *Pacific Graphics*, pages 53–57, 2015. 1
- [47] Kangxue Yin, Zhiqin Chen, Siddhartha Chaudhuri, Matthew Fisher, Vladimir G Kim, and Hao Zhang. Coalesce: Component assembly by learning to synthesize connections. In *International Conference on 3D Vision (3DV)*, pages 61–70, 2020. 2
- [48] Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas J Guibas, Hao Dong, et al. Generative 3d part assembly via dynamic graph learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 6315–6326, 2020. 2
- [49] Fanxing Zhang, Zhihao Liu, Zhanglin Cheng, Oliver Deussen, Baoquan Chen, and Yunhai Wang. Mid-air finger sketching for tree modeling. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 826–834, 2021. 2
- [50] Chuhan Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 900–909, 2017. 2