Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

VegMRFP: Mixed Radiance Field Primitives for Vegetation Rendering and Reconstruction

Anonymous Authors

ARTICLE INFO

Article history:

Keywords: 3D Reconstruction, Shape Modeling, Vegetation, Radiance Field

ABSTRACT

Vegetation constitutes one of the most abundant forms of life on Earth, yet accurately representing and reconstructing vegetation using radiance fields with low storage overhead remains a significant challenge. This difficulty arises from the intricate geometry of vegetation, characterized by numerous thin twigs and delicate leaves. To tackle this problem, we introduce VegMRFP, a novel framework for vegetation rendering and surface reconstruction based on mixed radiance primitives. Our method employs 2D Gaussian primitives to model plant stems and billboard primitives to represent clusters of leaves and lobes. Combined with a storage optimization strategy, the framework enables high-quality rendering of vegetation with lower memory cost. Experimental results demonstrate that our method achieves vegetation rendering quality comparable to the state-of-the-art methods, while requiring only 61.3% and 71.2% of the memory cost of BBSplat and 2DGS, respectively. Project page: `https://luminarygrove315.github.io/VegMRFP/`.

1. Introduction

Vegetation accounts for approximately 82% of the Earth's biomass [1], making them indispensable in various application fields, from video games and film production to urban planning, architectural design, and forestry simulation. However, due to the fractal characteristics of most vegetation and the numerous intricate details of leaves and twigs, reconstructing photorealistic 3D vegetation models remains a challenging task. While classical vegetation modeling methods can reconstruct vegetation with clear hierarchical structures, the process is laborious and the results often lack realism. Billboard technique [2–6] serves as a computationally efficient approach for vegetation rendering in real-time applications. It uses dynamically oriented 2D texture patches (billboards) in place of complex 3D geometry, continuously adjusting their facing direction relative to the camera orientation to maintain visual fidelity. Therefore, billboards offer significant advantages for vegetation rendering including superior rendering performance, reduced memory consumption, and inherent support for dynamic visual effects.

In recent years, Neural Radiance Fields (NeRF) [7] and 3D Gaussian Splatting (3DGS) [8] have emerged as a prominent re-

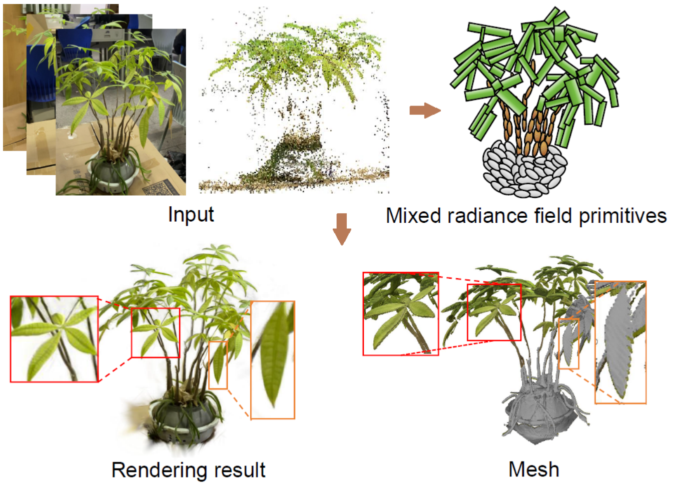

Fig. 1. VegMRFP employs 2D Gaussian primitives to model vegetation stems, while utilizing billboard primitives to represent lobe clusters such as leaves and twigs.

search focus within the domains of novel view synthesis and 3D reconstruction, due to their high fidelity and strong scalability. For vegetation reconstruction, although NeRF and 3DGS-based methods demonstrate significant advantages for producing realistic models in overall morphology, they exhibit three major limitations: (a) reconstruction quality deteriorates in heavily self-occluded leaf and twig regions; (b) requiring extensive training time; (c) the generated models requires substantial storages.

More recently, Huang et al. [9] proposed a novel 2D Gaussian Splatting (2DGS) approach to model and reconstruct geometrically accurate radiance fields from multi-view images. Unlike 3DGS, 2DGS provides view-consistent geometry while modeling surfaces intrinsically, allowing for noise-free and detailed geometry reconstruction, fast training speed, and real-time rendering. BBSplat [10] is a similar approach for novel view synthesis based on textured geometric primitives. The proposed primitives narrowed the rendering quality gap between 2D and 3D Gaussian Splatting, enabling the accurate extraction of 3D mesh as in the 2DGS framework.

The crown of vegetation can be roughly divided into the stems (or trunk and main branches) and the texture-lobes [11], and the latter contain intricate details of leaves and twigs. Inspired by the aforementioned works, we present VegMRFP, a mixed radiance-field primitive framework for storage-efficient 3D vegetation rendering and surface reconstruction that combines 2DGS and BBSplat with inherent vegetation characteristics as shown in Figure 1. The approach takes multi-view vegetation images with registered camera poses as input, constructs realistic 3D representations through mixed radiance field primitives, and ultimately extracts geometrically accurate mesh models of the vegetation. Specifically, we implement a two-stage strategy for vegetation radiance field representation. In the initial stage, 2D Gaussian primitives are employed to coarsely represent the complete vegetation structure, and the DBSCAN algorithm[12] is applied to separate lobe clusters from the 2D Gaussian primitive clusters. The second stage uses differentiated 2D Gaussian primitives for less-detailed stem structures, while billboard primitives handle the heavily self-occluded, detailed lobe clusters. Although the mixed primitive representation enhances rendering quality, it significantly increases the storage requirements of the resulting radiance field models. Thus, we developed an optimization method to minimize storage consumption while preserving rendering fidelity to the greatest possibility. In summary, our research makes the following contributions:

- A two-stage mixed representation framework VegMRFP for realistic vegetation rendering and surface reconstruction. This framework employs 2D Gaussian primitives for stem structures and billboard primitives for lobe cluster representation, ultimately enabling high-fidelity 3D mesh extraction of complete vegetation models.
- The memory usage of VegMRFP is reduced via Spatial Redundancy Scoring (SRS) and variable Spherical Harmonics (SH) order, enabling high-quality vegetation rendering with lower storage cost.
- A multi-view vegetation dataset is constructed and four metrics are used for facilitating quantitative evaluation of reconstruction quality of various vegetation.

2. Related Work

This study is closely aligned with three primary research areas: 3D vegetation reconstruction, radiance field representation, and radiance field-based surface reconstruction. Accordingly, this review focuses primarily on these domains.

2.1. *3D Vegetation Reconstruction*

Vegetation reconstruction has remained a prominent cross-disciplinary research topic in computer graphics and botanical sciences. Early studies mainly relied on rule-based or procedural branching models [13], which can generate structured plant architectures but often require hand-crafted rules and parameters. TreeSketchNet [14] framework incorporates deep learning to generate high-fidelity vegetation skeletons from sketch inputs, significantly advancing reconstruction quality.

In addition to the aforementioned reconstruction approaches, leveraging both point clouds and images as model inputs often yields 3D vegetation models with significantly enhanced photorealism. Early point cloud-based vegetation reconstruction methods [15–17] were often constrained by multiple manual parameter adjustments, unnatural reconstruction outcomes, and relatively slow computational performance. The work in [18] introduced a parameter-free approach for vegetation reconstruction from non-uniform laser-scanned point clouds. AdTree[19] automatically reconstructs high-fidelity 3D vegetation skeletons from laser-scanned point clouds using its three-module reconstruction pipeline. SmartTree[20] employs deep learning to extract vegetation skeletons from point clouds, achieving superior performance on quantitative metrics. Image-based vegetation 3D reconstruction methods [21–25] compute depth maps, voxels, and point clouds from input images, then extract skeletal structures through specialized algorithms before generating final 3D mesh models using L-Systems. Li et al.[26] applies deep learning to single-image vegetation reconstruction, employing radial boundary volumes to produce photorealistic 3D vegetation models from individual photographs.

2.2. *Radiance Field Representation*

NeRF[7] was initially proposed for novel view synthesis, implicitly representing scenes as a continuous 5D function and using volume rendering to guide 3D reconstruction via 2D images. However, its early versions suffered from low rendering quality and slow training. To improve rendering fidelity, researchers enhanced NeRF through better input encoding [27], sampling strategies [28], and scene representations [29, 30]. Liu et al. and Hedman et al. [31, 32] accelerated rendering using sparse voxel octrees, while Instant-NGP [33] replaced positional encoding with multi-resolution hash encoding and optimized CUDA implementation, achieving orders-of-magnitude speedups without quality loss.

3DGS [8] has emerged as a prominent novel view synthesis method in recent years. To enhance its rendering quality, MSGS [34] uses a multi-scale 3D Gaussian splatting algorithm that maintains Gaussian distributions at different scales to represent the same scene. Mip-Splatting [35] employs a 3D smoothing filter constrained by the maximum sampling frequency from input views to regulate Gaussian primitive sizes, effectively eliminating high-frequency artifacts during magnification. Rota et al. However, as image scales and resolutions increase, the growing number of Gaussian primitives and their storage demands pose challenges. Real-time rendering in large-scale or high-resolution scenarios is constrained by limited computational and memory resources. To address this problem, Light-Gaussian [36] employs pruning strategies to remove redundant primitives. Compact-3DGS [37] proposes a learnable masking strategy. Additionally, R-3DGS [38] analyzes the primary factors behind 3DGS's high memory consumption primitives count and proposes three compression schemes to reduce model storage overhead.

2.3. *Radiance Field-Based Surface Reconstruction*

Recently, neural network-based level-set methods [39–41] for representing 3D shapes have gained popularity. These approaches can be integrated with radiance field techniques by representing a 3D surface as the zero level-set of a Signed Distance Field (SDF) [42–44], followed by mesh extraction algorithms to obtain the final 3D model. Building on NeuS [42], NeuS2[45] replaced the original NeRF with Instant-NGP, enabling faster and higher-quality reconstructions for both static and dynamic scenes. Similarly, Neuralangelo [46] integrated Instant-NGP with SDF by employing numerical differentiation to solve second-order derivative issues and coarse-to-fine optimization to ensure stability, ultimately achieving high-resolution reconstructions for indoor and outdoor scenes, though its multi-stage upsampling process increased training time and resource consumption. In the domain of explicit Gaussian primitive representation for high-fidelity 3D mesh reconstruction, SuGar [47] introduces a regularization term that aligns Gaussian points with scene surfaces. Meanwhile, 2DGS [9] employs 2D Gaussian disks to represent 3D scenes while ensuring multi-view consistency through depth warping and normal consistency constraints, thereby improving reconstruction quality.

Since vegetation has complex geometry structures, the general radiance field-based rendering and reconstruction methods will result in either high storage consumption or poor rendering quality. This motivates our low storage-cost and high-fidelity approach combining mixed radiance field primitives for vegetation-specific reconstruction.

3. Preliminaries

3.1. *2D Gaussian Splatting*

3D Gaussian Splatting (3DGS) [8] explicitly represents 3D scenes using collections of Gaussian primitives. Unlike 3DGS which employs 3D ellipsoids, 2DGS represents scenes using flat 2D ellipses, which are aligned with object surfaces and exhibit Gaussian-distributed transparency. The 2D ellipses are parameterized as $\{\mu_i, s_i, r_i, o_i, SH_i\}$, denoting their position, 2D scale, quaternion-based rotation, global opacity, and Spherical Harmonics (SH) which encode view-dependent color properties. To determine ellipse orientation, 2DGS leverages local tangent planes, which are derived from a transformation matrix H:

$$P(u, v) = \mathbf{p}_k + s_u\mathbf{t}_u u + s_v\mathbf{t}_v v = H(u, v, 1, 1)^T \tag{1}$$

$$H = \begin{bmatrix} s_u\mathbf{t}_u & s_v\mathbf{t}_v & \mathbf{0} & \mathbf{p}_k \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{RS} & \mathbf{p}_k \\ \mathbf{0} & 1 \end{bmatrix} \tag{2}$$

Where the (u, v) coordinates lie on the local tangent plane P. Using the coordinates, the 2D Gaussian intensity is denoted by:

$$G(\mathbf{u}) = e^{-\frac{u^2+v^2}{2}} \tag{3}$$

3.2. *Billboard Splatting*

BillBoard Splatting (BBSplat) is an optimized variant of 2DGS that replaces the volumetric density in 2D Gaussian primitives with an α texture T^{α}. For color encoding, the method further incorporates an RGB texture T^{RGB} to compute the color:

$$c(x) = \sum_{i=1} c_i[\mathbf{u}(x)]T_i^{\alpha}[\mathbf{u}(x)] \prod_{j=1}^{i-1}(1 - T_j^{\alpha}[\mathbf{u}(x)]) \tag{4}$$

$$c_i[\mathbf{u}] = T_i^{RGB}[\mathbf{u}] + RGB(SH_i, \mathbf{d_i}) \tag{5}$$

Where function $\mathbf{u}(\cdot)$ computes texture coordinates; $RGB(SH_i, \mathbf{d_i})$ represents the color value calculated using the SH features.

4. Methods

Figure 2 shows the workflow of the VegMRFP framework. Starting from multi-view vegetation images and an initial point cloud, we first optimize 2D Gaussian primitives to establish a preliminary radiance field representation of the complete vegetation (Sec. 4.1). Then, we compute SRS for each optimized 2D Gaussian, and employ DBSCAN clustering based on both SH features and SRS to segment stems from lobe clusters (Sec. 4.2). In the second stage, depth distortion and normal consistency losses are applied to the refined 2D Gaussian primitives for stems to enhance geometric consistency for TSDF-based mesh extraction. Lobe clusters are represented by billboard primitives with learnable RGB and α textures and are optimized through photometric supervision. (Sec. 4.3). Finally, we introduce a storage optimization method to minimize storage consumption of the vegetation model (Sec. 4.4).

4.1. *Initial Radiance Field Representation*

We first employ 2D Gaussian primitives to provide an initial representation of the entire vegetation, ensuring multi-view consistency while facilitating subsequent extension with mixed primitives. The photometric loss, which consists of a RGB reconstruction term $\mathcal{L}_1$ and a Structural Similarity Index Measure (SSIM) term $\mathcal{L}_{D-SSIM}$, is used for the intial training:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM} \tag{6}$$

Where λ is a hyperparameter.

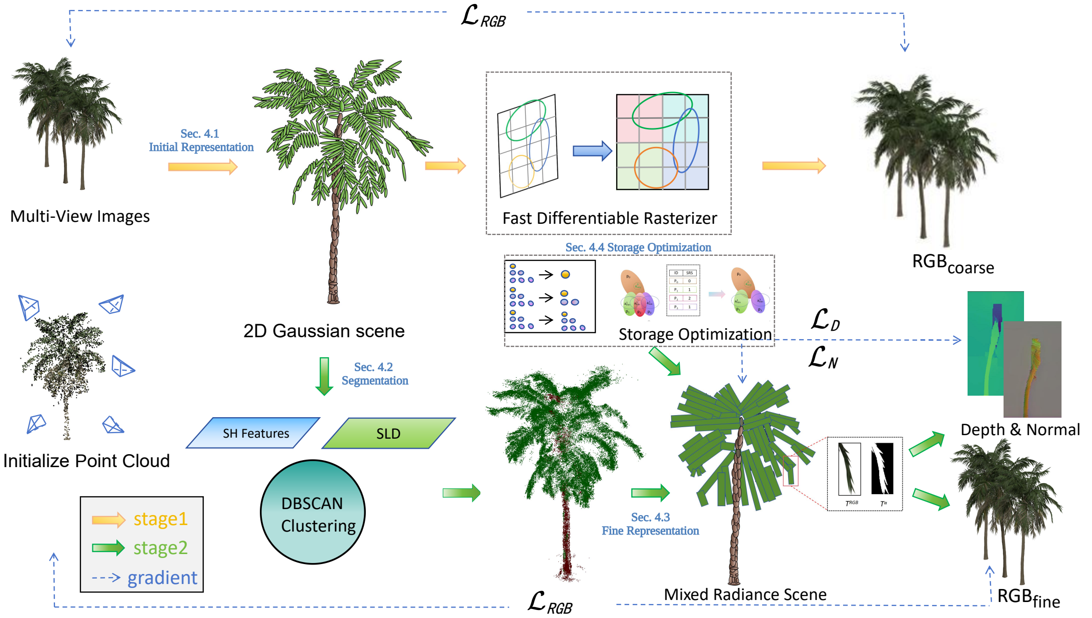

Fig. 2. Overview of the VegMRFP framework.

4.2. Segmentation of Stems and Lobe Clusters

As empirically observed, vegetation stems (trunk and main stems) and lobe clusters (leaves and twigs) exhibit significant disparities in both chromatic attributes and spatial density distributions. Motivated by this prior, we propose a segmentation method that jointly leverages spherical harmonic (SH) coefficients and Spatial Local Density (SLD) metrics as shown in Figure 3.

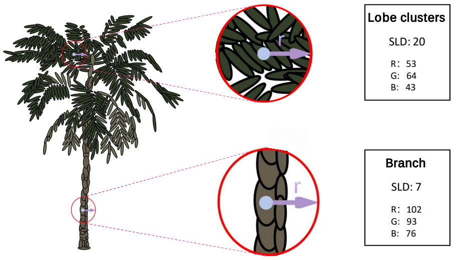

Fig. 3. Vegetation lobe clusters and stems exhibit distinct characteristics in both RGB color space and local spatial density distributions. Specifically, lobe clusters exhibit significantly higher SLD of Gaussian primitives with dominant coloration. In contrast, stems demonstrate sparser primitive distributions and characteristic hues consistent with its woody tissue reflectance properties.

Calculation of SLD. For each 2D Gaussian primitive with center $\mathbf{p}_i$, we select its k nearest neighboring primitives with centers $\{\mathbf{p}_{ij}\}_{j=1}^{k}$. Let $\mathbf{d}_{ij} = \mathbf{p}_{ij} - \mathbf{p}_i$ denote the displacement vector from the target primitive to its j-th nearest neighbor. The Spatial Local Density (SLD) is then defined as the inverse of the average distance to the k nearest neighbors:

$$\rho_i = \frac{k}{\sum_{j=1}^{k} \|\mathbf{d}_{ij}\|_2}. \tag{7}$$

Intuitively, this KNN-based SLD measures the local density of neighboring primitives around the target primitive. A smaller average distance to the k nearest neighbors leads to a larger ρ_i, indicating a denser local primitive distribution. This can be intuitively understood as counting the number of 2D Gaussian primitives within a spherical region centered at the primitive's centroid with a specified radius, where a higher count indicates greater density and vice versa. We compute the SLD of each 2D Gaussian primitive using a standard *KNN* algorithm, implemented with CUDA programming through Morton codes and reduction operations to efficiently find k nearest neighbors and calculate the sum of distances between their centers. In our experiments, we set k as 16.

DBSCAN for segmenting stems and lobe clusters. DBSCAN, a classical density-based clustering algorithm capable of forming arbitrarily shaped clusters while identifying noise points, is employed to seperate stems and lobe clusters. We define the composite density metric σ_{ds} as a weighted sum of n SH coefficients and the computed SLD is denoted by:

$$\sigma_{ds} = \frac{w}{n} \cdot \sum_{i=1}^{n} SH_i + (1 - w) \cdot \rho \tag{8}$$

Where we set the weight parameter w to0.4 by considering the greater impact of spatial local density on the clustering algorithm, and configure the n as 48. In the initial 2D Gaussian primitive representation of vegetation, each specimen typically contains approximately $100,000$ primitives, with stem components accounting for roughly 20% of the total. Based on this distribution characteristics, we set the minimum points parameter ($minPts$) to $20,000$ for the DBSCAN algorithm. After completing the clustering process, we store the segmentation results in an additional semantic parameter cl defined for each 2D Gaussian primitive. The parameter is assigned binary value, where $cl = 1$ indicates stem primitive and $cl = 0$ represents lobe cluster primitive.

4.3. Fine Representation of Vegetation

We observe that even with a large number of 2D Gaussian primitives, accurately representing vegetation lobe clusters for high-quality rendering remains challenging. Motivated by the classical billboard technique for efficient vegetation rendering [3, 4, 48] and the BBSplat approach for 3D scene representation, we propose a mixed representation of vegetation by combining 2D Gaussian primitives with billboard elements.

Fine representation of 2DGS stems. Since stems exhibit relatively sparse geometric details and smooth surfaces, we employ 3DGS's adaptive density control strategy to refine their representation as shown in Figure 4. First, we clone 2D Gaussian primitives in under-reconstructed areas while splitting them in over-reconstructed regions. Then, we prune primitives with opacity below 0.05 to prevent excessive counts during every 3,000 iterations. The optimization further incorporates distortion loss and normal consistency loss to enhance stem geometric details, enabling extraction of high-quality 3D mesh model.

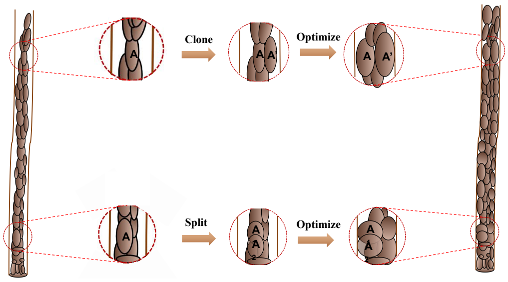

Fig. 4. Cloning and splitting operations for refining the representation of stems using 2D Gaussian primitives.

BillBoard primitive-based lobe cluster representation. Unlike the stems, lobe clusters have many small twigs and leaf details, making the representation quality of 2D Gaussian primitives relatively poor. Therefore, we use billboard primitives instead of 2D Gaussian primitives to optimize the lobe cluster representation as shown in Figure 5. Specifically, we extract the primitives with $cl = 0$ from the initial 2D Gaussian representation, then assign each primitive a 16×16 α texture and an 16×16 RGB texture. These textures are optimized using a texture loss term $\mathcal{L}_{texture}$, formulated as follows:

$$\begin{aligned}
\mathcal{L}_{rgb} &= \frac{1}{N}\sum_{i=0}^{N} w_i \| T_i^{RGB} \|, \\
\mathcal{L}_{\alpha} &= \frac{1}{N}\sum_{i=0}^{N} w_i \| T_i^{\alpha} - \mathcal{G} \|, \qquad (9) \\
\mathcal{L}_{texture} &= \lambda_{rgb}\mathcal{L}_{rgb} + \mathcal{L}_{\alpha}
\end{aligned}$$

Where N denotes the total number of Billboard primitives, w_i represents the visibility weight of the i-th primitive, and $\mathcal{G}$ denotes the Gaussian distribution.

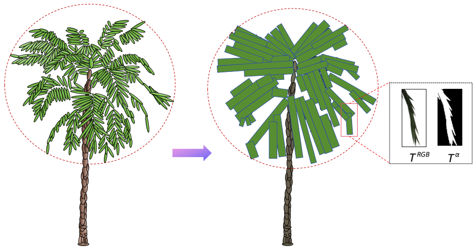

Fig. 5. Representation of vegetation lobe clusters using the billboard primitives for further enhancing the rendering quality.

4.4. Storage Optimization

While our method improves the rendering quality of individual vegetation, it also results in a larger storage footprint for the final vegetation model. Inspired by the work of R-3DGS [38], we introduce a storage optimization method by considering spatial redundancy scoring and variable spherical harmonics order of the generated primitives. Different from R-3DGS, which focuses on compressing standard 3D Gaussian primitives, our storage optimization is adapted to the proposed mixed-primitives representation consisting of 2D Gaussian primitives and billboard primitives.

Spatial redundancy scoring (SRS). SRS is defined as the number of primitives that overlap with a small sphere centered at primitive p with a radius r. From multiple viewpoints, we select the view with the highest resolution where primitive p is observed. The dimension of each splat is considered through its spatial influence region in this highest-resolution view, which is determined by the projected extent of the corresponding 2D Gaussian or billboard primitive. Then, we use the KNN algorithm to find a series of nearest neighboring primitives of p in 3D space. For each of these candidate primitives, we calculate the spatial area they influence when viewed from p's highest-resolution perspective. Next, we count how many of these areas intersect with p's bounding sphere as p's SRS. After calculating the SRS for all primitives, we compute the mean μ and standard deviation σ of all SRS values, and then sort all primitives by their SRS to optimize pruning efficiency. Finally, we set adaptive threshold as $\tau_p = \max(\mu + \lambda_r \sigma, 3)$, and remove all primitives with SRS values exceeding this threshold. This process can be illustrated in Figure 6.

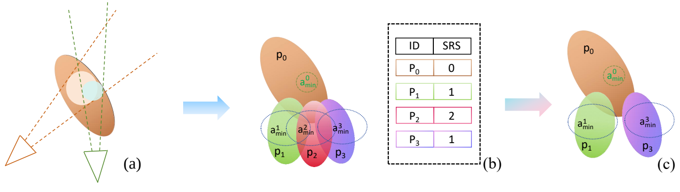

Fig. 6. Illustration of storage optimization process by considering spatial redundancy scoring. (a) For each primitive, identify the viewpoint where it appears at maximum resolution. (b) In a given viewpoint, each primitive p_i covers a spherical region a^i_{min}. Primitive p_0's spherical region intersects with none of the others so its $SRS = 0$. Primitives p_1, p_2, p_3 intersect with 1, 2, and 1 spherical regions respectively, thus their $SRS = 1, 2, 1$. (c) Since primitive p_2 intersects with both p_1 and p_3 and its redundancy score ($SRS = 2$) exceeds a threshold, p_2 gets removed.

Table 1. PSNR(↑) of 9 models on 16 types of vegetation

Vegetation	INGP	3DGS	BBSplat	R-3DGS	Neus2	2DGS	SuGaR	G-surfels	ours
Virtual Pine	35.115	36.393	37.909	35.521	36.093	36.082	35.617	35.503	37.012
Virtual Palm	28.746	29.107	29.809	29.002	28.756	29.111	28.553	29.202	29.401
Virtual Bare Tree	28.714	29.351	30.212	29.140	28.661	29.132	28.221	29.203	30.643
Virtual Pear	28.252	31.022	29.189	30.643	29.351	29.668	23.212	29.132	29.203
Virtual Ulmus	27.217	27.125	29.018	26.728	26.999	27.818	27.565	28.010	28.131
Virtual Sunflower	38.312	38.544	40.138	38.244	38.123	38.983	37.969	39.213	39.212
Virtual Cucumber	35.407	38.081	38.374	37.877	37.902	37.233	34.121	38.203	37.989
Virtual Aloe	41.960	42.033	41.995	41.822	41.898	41.911	40.937	41.943	41.952
Bare Tree	23.421	24.456	26.123	24.322	23.111	24.567	23.654	23.987	25.789
Locust	22.656	23.789	23.987	22.567	21.321	22.456	21.987	23.112	23.436
Crape Myrtle	22.654	24.987	24.123	23.456	22.322	23.567	22.114	24.432	23.789
Pine	20.987	22.345	22.876	21.123	20.456	21.234	20.654	21.789	22.111
Maple	20.876	22.765	22.431	22.512	20.654	21.123	20.987	21.234	22.010
Euo Jap	22.345	24.987	23.947	24.123	22.113	23.567	22.654	24.323	24.003
Shurb	25.317	27.582	27.916	26.209	25.746	26.438	25.893	26.125	27.369
Potting	24.317	26.393	26.914	25.218	24.766	25.441	24.901	25.124	26.582
Mean	27.894	29.310	29.685	28.657	28.017	28.646	27.440	28.783	29.290

Variable Spherical Harmonics (SH) Order. Both 2D Gaussian primitives and billboard primitives typically use third-order SH for view-dependent color encoding. However, many primitives can achieve equal high-quality rendering result with lower-order SH representations. Thus, we implement a two-tiered approach for SH order reduction. First, for primitives with view-independent colors, we encode their color using only 0th-order SH coefficients. On the other hand, we calculate the Euclidean distances d_0, d_1, d_2 between the RGB colors encoded with 0th-order, 1st-order, 2nd-order SH coefficients and those encoded with full 3rd-order SH coefficients respectively. These distances are then weighted by the average opacity across different viewpoints to obtain the final weighted average distance. If any d_i falls below a threshold ϵ_σ, it indicates that the primitive's color can be sufficiently encoded using the i-th order SH coefficients (where i ranges from 0 to 2). This process dynamically determines the minimal required SH order for each primitive while maintaining rendering quality.

4.5. *Mesh Extraction*

For vegetation stems, we compute a Truncated Signed Distance Field (TSDF) using the median depth and normal vectors obtained from the optimized 2D Gaussian primitives, then extract the 3D mesh via the classical Marching Cubes (MC) algorithm. For vegetation lobe clusters, we triangulate the billboard primitives and store the vertex color values of the resulting triangular patches in texture maps.

5. Results

5.1. *Datasets and Evaluation Metrics*

Datasets. To verify the effectiveness of the proposed methods for individual vegetation, we captured multi-view images of 8 real vegetation species, and the image background was removed through the large-scale segmentation model SAM[49]. In addition, 8 virtual vegetation models were created using SpeedTree software. Radiance field vegetation datasets were constructed from both the real-world and the virutal models. The number of images for each individual vegetation ranges from 100 to 150, with a pixel resolution of 1920×1080.

Evaluation Metrics. To facilitate a quantitative evaluation of the rendering quality, we adopted Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSIM) and Learned Perceptual Image Patch Similarity (LPIPS) metrics. On the other hand, Chamfer Distance (CD) was employed to assess the reconstruction quality across all test vegetation samples.

Implementation Details. Our method comprises two stages with three key components. In the initial vegetation representation stage, we train 2D Gaussians for only $7,000$ iterations using solely photometric loss for optimization. For the vegetation segmentation component, we configure the DBSCAN algorithm with a neighborhood radius ϵ of 0.05 and minimum points (*minPts*) parameter of $20,000$. In the radiance field refinement stage, we train 2D Gaussian primitives for $30,000$ iterations with depth distortion loss weight α set to $1,000$ and normal consistency loss weight β to 0.05. For billboard primitives, the training runs for $32,000$ iterations with texture loss weights $\lambda_{rgb} = \lambda_\alpha = 1e^{-4}$ and a maximum primitive count of $120,000$. During mesh extraction, we set the voxel size as 0.004. All baseline methods maintain their original paper configurations for consistency, and all experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU.

Parameter influence and trade-offs. We further discuss the influence of key parameters. Higher image resolution and more input views generally improve fine-detail coverage, but increase training cost and may require more primitives. For SLD computation, the neighbor number controls the locality of density

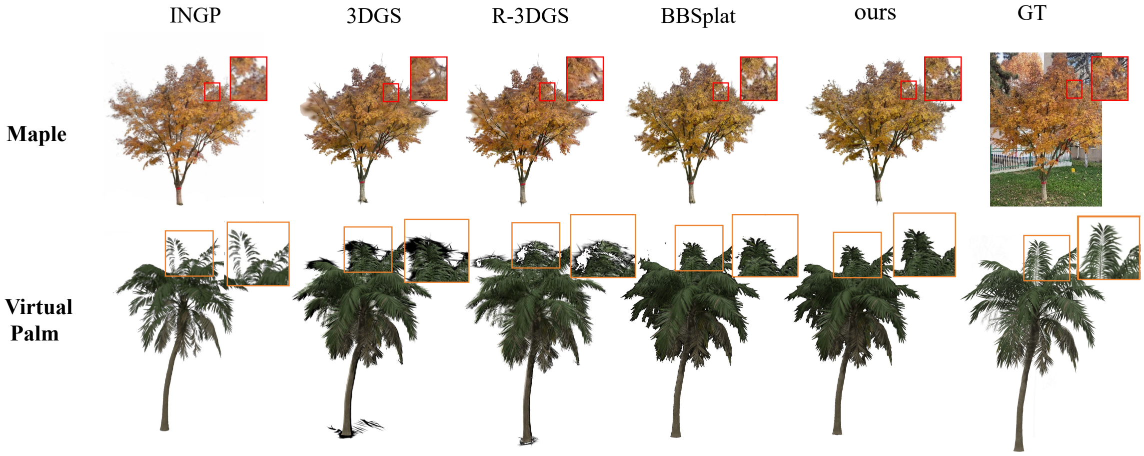

Fig. 7. Comparison of rendering quality with rendering-oriented radiance fields on two types of vegetation.

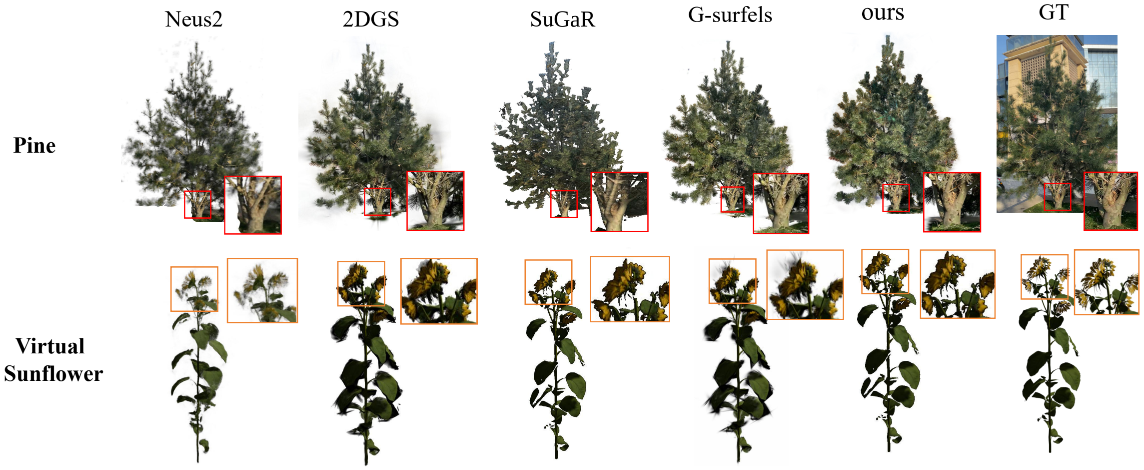

Fig. 8. Comparison of rendering quality with reconstruction-oriented radiance fields on two types of vegetation.

estimation: a small k may be sensitive to local noise, while a large k may over-smooth the density difference between stems and lobe clusters. We empirically set $k = 16$. For DBSCAN, ϵ and minPts affect the stem/lobe partition; overly small values may cause over-segmentation, whereas overly large values may merge different structures. We use $\epsilon = 0.05$ and minPts = 20,000 across all experiments. Training iterations and loss weights balance visual fidelity, geometric consistency, and training time. For SRS, using more views or neighboring candidates may improve redundancy estimation but increases computational cost; therefore, we use the highest-resolution view and a KNN-based local neighborhood for efficient pruning.

5.2. Evaluation of Rendering Quality

In our experiments, we categorized radiance field methods into two distinct classes for comparative analysis of rendering quality. One is **rendering-oriented radiance fields**, which primarily focuses on color and lighting accuracy without strict surface alignment constraints, typically achieving superior rendering fidelity. The evaluated approaches include Instant-NGP (INGP) [33], 3DGS [8], BBSplat [10] and R-3DGS [38]. The other is surface **reconstruction-oriented radiance fields**, which incorporates additional regularization terms to ensure geometric surface alignment, often at the cost of slight rendering precision. The compared algorithms includes Neus2 [45], 2DGS [9], SuGaR [47] and G-surfels [50].

Figure 7 and 8 show the comparative results generated by the two kinds of radiance field methods. The results reveal that our approach maintains comparable rendering quality to the state-of-the-art **rendering-oriented radiance fields**, while consistently outperforming **reconstruction-oriented radiance fields** in terms of visual fidelity such as fine botanical features and natural illumination characteristics. Table 1 shows quantitative results of PSNR metric of the two kinds of radiance field methods. The average PSNR of our method on 16 types of vegetation is 29.290, which demonstrates comparable rendering quality to **rendering-oriented radiance fields** such as 3DGS and BBSplat, and significant rendering advantages over **reconstruction-oriented radiance fields**. The results of SSIM and LPIPS metrics show consistency with PSNR-based results, and the average SSIM and LPIPS metric values of our method are 0.872 and 0.163 respectively, achieving optimal or suboptimal values on most vegetation datasets. Detailed per-scene comparisons are provided in Tables 2 and 3.

Table 2. SSIM(↑) of 9 models on 16 types of vegetation

Vegetation	INGP	3DGS	BBSplat	R-3DGS	Neus2	2DGS	SuGaR	G-surfels	ours
Virtual Pine	0.917	0.946	0.957	0.935	0.897	0.933	0.926	0.930	0.949
Virtual Palm	0.855	0.875	0.901	0.869	0.858	0.876	0.886	0.893	0.902
Virtual Withered	0.893	0.935	0.932	0.927	0.886	0.919	0.907	0.911	0.936
Virtual Pear	0.885	0.920	0.899	0.903	0.897	0.900	0.851	0.892	0.894
Virtual Ulmus	0.852	0.883	0.917	0.876	0.849	0.877	0.861	0.869	0.917
Virtual Pear	0.941	0.960	0.964	0.957	0.939	0.958	0.942	0.962	0.963
Virtual Cucumber	0.913	0.958	0.962	0.949	0.899	0.927	0.919	0.957	0.959
Virtual Aloe	0.963	0.972	0.968	0.969	0.965	0.968	0.964	0.970	0.967
Withered	0.784	0.812	0.859	0.721	0.689	0.732	0.798	0.814	0.843
Locust	0.753	0.803	0.847	0.698	0.671	0.714	0.782	0.769	0.829
Crape Myrtle	0.769	0.851	0.822	0.831	0.683	0.726	0.791	0.837	0.804
Pine	0.718	0.715	0.719	0.689	0.671	0.698	0.709	0.706	0.712
Maple	0.721	0.712	0.723	0.692	0.674	0.701	0.705	0.706	0.718
Euo Jap	0.771	0.853	0.828	0.715	0.688	0.729	0.793	0.839	0.815
Shurb	0.841	0.883	0.891	0.879	0.769	0.812	0.856	0.874	0.876
Potting	0.832	0.861	0.879	0.789	0.758	0.801	0.847	0.859	0.871
Mean	0.838	0.871	0.879	0.837	0.799	0.830	0.846	0.862	0.872

5.3. *Evaluation of Storage and efficiency Consumption*

The quantitative comparison results with explicit radiance field representations are shown in Table 4. Across 16 vegetation datasets, our method achieves an average storage footprint of 124.8MB, consuming just 61.3% of the baseline BBSplat's storage requirements (203.6MB) and 71.2% of the 2DGS's memory cost (175.3MB) in storage efficiency. The result demonstrates that our storage optimization method by combining SRS and variable SH order has obvious advantage in compact vegetation radiance field representation.

To further evaluate the efficiency of VegMRFP, we additionally report the training time comparison in Table 7. The DBSCAN-based stem/lobe segmentation is accelerated with CUDA and takes less than one minute in our experiments. Therefore, the total training time is mainly determined by the initial 2DGS training stage and the billboard refinement stage. On average, VegMRFP takes 55.56 minutes on the 16 vegetation datasets, which is close to 2DGS (53.19 minutes) and about 27.1% faster than BBSplat (76.19 minutes). Combined with the storage-consumption results, these results show that VegMRFP achieves competitive training efficiency while maintaining a lower memory footprint.

Ablation study. We conducted ablation study to evaluate the impact of both SRS and variable SH order on storage requirements and rendering quality, and the experimental results are shown in Table 5. The results demonstrate that both SRS and SH can effectively reduce final model size, while SRS additionally improving rendering quality while achieving storage reduction. SH significantly decreases storage costs by reducing color feature encoding overhead, though at a modest compromise to visual fidelity.

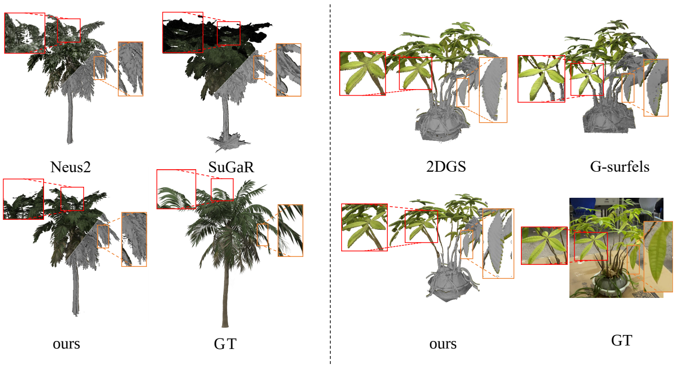

Fig. 9. Qualitative comparison of reconstruction quality with baselines on two types of vegetation.

5.4. *Evaluation of Surface Reconstruction Quality*

Finally, we compared our method with 4 radiance field models for 3D vegetation surface reconstruction across 16 types of vegetation using the standard CD metric as shown in Table 6. To ensure scale consistency among output meshes from different models, we first aligned all reconstructed meshes to our manually crafted reference vegetation models, then normalized all vertex coordinates to the range $[0, 1]$. Using Poisson disk sampling, we generated $16,000$ points from each mesh for CD calculation. Quantitative evaluation result shows that after vertex normalization, our method reduces the average CD by 0.002 compared to the previous state-of-the-art surface reconstruction approach G-surfels, and by 0.05 relative to the 2DGS method. Figure 9 shows that the reconstructed mesh of our method captures better botanical features including leaf venation patterns, serrated edges, and intricate geometric textures than Neus2, SuGaR, 2DGS and G-surfels. Both the quantitative and qualitative results validate the capability of our method to gener-

Table 3. LPIPS(↓) of 9 models on 16 types of vegetation

Vegetation	INGP	3DGS	BBSplat	R-3DGS	Neus2	2DGS	SuGaR	G-surfels	ours
Virtual Pine	0.051	0.049	0.040	0.057	0.052	0.051	0.063	0.054	0.043
Virtual Palm	0.135	0.118	0.098	0.121	0.143	0.122	0.119	0.104	0.115
Virtual Withered	0.075	0.064	0.068	0.069	0.081	0.072	0.082	0.071	0.065
Virtual Pear	0.102	0.077	0.094	0.081	0.104	0.113	0.086	0.097	0.095
Virtual Ulmus	0.124	0.112	0.089	0.117	0.127	0.113	0.136	0.109	0.111
Virtual Sunflower	0.031	0.025	0.024	0.026	0.033	0.028	0.041	0.027	0.028
Virtual Cucumber	0.045	0.034	0.037	0.042	0.051	0.036	0.055	0.033	0.032
Virtual Aloe	0.024	0.019	0.021	0.022	0.026	0.022	0.027	0.023	0.020
Withered	0.314	0.233	0.212	0.241	0.323	0.246	0.301	0.221	0.215
Locust	0.382	0.288	0.295	0.301	0.379	0.291	0.377	0.312	0.297
Crape Myrtle	0.391	0.265	0.277	0.281	0.356	0.299	0.384	0.301	0.279
Pine	0.484	0.336	0.315	0.376	0.491	0.378	0.473	0.335	0.344
Maple	0.474	0.349	0.311	0.361	0.483	0.401	0.464	0.363	0.323
Euo Jap	0.331	0.276	0.268	0.294	0.354	0.371	0.388	0.313	0.273
Shurb	0.292	0.184	0.167	0.191	0.276	0.186	0.211	0.201	0.199
Potting	0.304	0.163	0.181	0.214	0.311	0.210	0.275	0.204	0.175
Mean	0.222	0.162	0.156	0.175	0.224	0.184	0.218	0.173	0.163

Table 4. Comparison of storage consumption of 5 models on 16 types of vegetation (MB)(↓)

Vegetation	2DGS	SuGaR	G-surfels	BBSplat	ours
Virtual Pine	142.8	634.6	147.8	160.8	**96.4**
Virtual Palm	143.0	634.1	144.9	172.9	**103.2**
Virtual Bare Tree	180.0	634.3	182.6	213.5	**149.1**
Virtual Pear	118.9	634.9	119.8	134.6	**75.3**
Virtual Ulmus	131.1	634.6	133.5	150.8	**90.1**
Virtual Sunflower	14.3	635.3	15.1	16.7	**10.1**
Virtual Cucumber	17.3	635.2	19.7	19.0	**13.3**
Virtual Aloe	15.2	635.4	16.4	18.5	**12.5**
Bare Tree	301.7	846.5	299.9	401.5	**240.9**
Locust	310.6	846.7	314.3	323.4	**157.0**
Crape Myrtle	312.5	847.9	322.5	313.2	**187.9**
Pine	212.7	847.4	224.8	267.1	**178.9**
Maple	303.4	846.3	302.1	366.8	**256.7**
Euo Jap	202.4	845.9	212.4	234.6	**140.7**
Shurb	199.5	843.2	188.7	227.4	**152.1**
Potting	198.7	844.9	201.3	237.5	**133.2**
Mean	175.3	740.5	177.9	203.6	**124.8**

ate higher-fidelity 3D vegetation mesh models with enhanced structural details.

To further validate the effectiveness of our geometry-aware optimization, we compare our reconstructed mesh with that generated by Screened Poisson Surface Reconstruction [51]. For fairness, we uniformly sampled 200,000 points from our final radiance field representation as input to Screened Poisson (implemented in Open3D with depth 9 and normal estimation). As shown in Figure 10, Screened Poisson produces severely incomplete surfaces with missing branches and over-smoothed foliage, failing to capture fine details due to its assumption of uniform point density and consistent normals–conditions unmet by vegetation's heterogeneous geometry. In contrast, our method faithfully preserves both the global morphological structure and fine local geometric details.

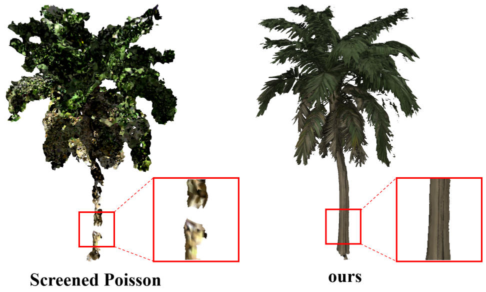

Fig. 10. Comparison of surface reconstruction results with Screened Poisson.

Figure 11 provides visual comparisons for six representative vegetation types, further illustrating the reconstruction performance across diverse species. As shown in the figure, our method recovers finer geometric details and more accurate surface structures, yielding results closer to the ground truth with fewer artifacts and higher fidelity.

6. Conclusions and Future Work

In this paper, we propose VegMRFP, a framework for vegetation rendering representation and surface reconstruction. Inspired by classical vegetation rendering and reconstruction techniques such as billboard and texture-lobes, we integrate radiance field by employing 2D Gaussian primitives for vegetation stems and billboard primitives for lobe clusters. Our framework incorporates both the spatial redundancy scoring and vari-

Table 5. Comparison of the effects of SRS and SH techniques on rendering quality and storage space (mean values of the vegetation datasets, w/o represents without)

Model	Virtual Palm				Real-world Maple			
	PSNR↑	SSIM↑	LPIPS↓	Mem. (MB)↓	PSNR↑	SSIM↑	LPIPS↓	Mem. (MB)↓
w/o SRS	29.845	0.938	0.071	172.3	21.638	0.692	0.345	299.7
w/o SH	31.906	0.876	0.059	199.3	22.435	0.721	0.319	343.4
w/o SRS&SH	**31.523**	**0.941**	**0.054**	222.5	**22.499**	**0.723**	**0.307**	386.4
Ours	30.643	0.936	0.065	**149.1**	22.010	0.718	0.323	**256.7**

Table 6. Quantitative evaluation result of surface reconstruction in virtual vegetation using metric CD (↓)

Virtual vegetation	Neus2	2DGS	SuGaR	G-surfels	ours
Pine	0.382	0.052	0.093	0.038	0.035
Palm	0.522	0.164	0.184	0.092	0.093
Withered	0.302	0.413	0.211	0.097	0.103
Pear	0.474	0.097	0.233	0.102	0.085
Ulmus	0.388	0.113	0.212	0.102	0.098
Sunflower	0.038	0.024	0.044	0.032	0.035
Cucumber	0.047	0.051	0.039	0.044	0.042
Aloe	0.031	0.033	0.036	0.035	0.037
Mean	0.273	0.118	0.132	0.068	0.066

Table 7. Comparison of training time of 5 methods on 16 types of vegetation (min)(↓)

Vegetation	2DGS	SuGaR	G-surfels	BBSplat	ours
Virtual Pine	43	165	52	64	**42**
Virtual Palm	**36**	164	57	66	44
Virtual Bare Tree	49	201	63	73	**47**
Virtual Pear	50	199	61	71	**49**
Virtual Ulmus	**47**	196	59	71	50
Virtual Sunflower	31	137	34	43	**29**
Virtual Cucumber	32	155	31	38	**27**
Virtual Aloe	**15**	96	27	33	19
Bare Tree	79	153	83	92	**67**
Locust	**70**	147	88	97	71
Crape Myrtle	77	162	85	101	**73**
Pine	76	155	84	97	**75**
Maple	**75**	154	86	99	77
Euo Jap	**68**	123	72	90	69
Shrub	**52**	149	87	91	72
Potting	**51**	132	79	93	78
Mean	**53.19**	155.5	65.5	76.19	55.56

able SH order to reduce storage requirements. Experimental results demonstrate that the proposed mixed radiance filed primitives achieve high-quality vegetation representation and surface reconstruction with lower memory consumption than previous methods.

However, our current study still has some limitations. The DBSCAN-based primitive partition is lightweight but heuristic, and may be sensitive to complex vegetation with ambiguous appearance and density patterns. Future work will explore more robust vegetation-specific or learning-based primitive segmentation methods. In addition, the current TSDF-based mesh extraction with classical Marching Cubes may produce unstable topology for thin or highly complex structures. We will investigate topology-aware surface extraction, adaptive TSDF resolution, and mesh repair post-processing. Finally, VegMRFP can be extended to a multi-component representation for stems, leaves, flowers, and rough bark regions, and further scaled to large vegetation communities.

References

[1] Bar-On, YM, Phillips, R, Milo, R. The biomass distribution on earth. Proceedings of the National Academy of Sciences 2018;115(25):6506–6511. doi:`10.1073/pnas.1711842115`.

[2] Décoret, X, Durand, F, Sillion, FX, Dorsey, J. Billboard clouds for extreme model simplification. In: ACM SIGGRAPH 2003 Papers. 2003, p. 689–696.

[3] Behrendt, S, Colditz, C, Franzke, O, Kopf, J, Deussen, O. Realistic real-time rendering of landscapes using billboard clouds. Computer Graphics Forum 2005;24(3):507–516. doi:`10.1111/j.1467-8659.2005.00876.x`.

[4] Garcia, I, Sbert, M, Szirmay-Kalos, L. Tree rendering with billboard clouds. In: Proceedings of Third Hungarian Conference on Computer Graphics and Geometry. 2005, p. 9–15.

[5] Bao, G, Zhang, X, Che, W, Jaeger, M. Billboards for tree simplification and real-time forest rendering. In: 2009 Third International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications. IEEE; 2009, p. 433–440.

[6] de Gomensoro Malheiros, M, Walter, M. A hybrid geometry and billboard-based model for trees. In: 2011 Brazilian Symposium on Games and Digital Entertainment. IEEE; 2011, p. 17–25.

[7] Mildenhall, B, Srinivasan, PP, Tancik, M, Barron, JT, Ramamoorthi, R, Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM 2021;65(1):99–106.

[8] Kerbl, B, Kopanas, G, Leimkühler, T, Drettakis, G. 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics(TOG) 2023;42(4):139–1.

[9] Huang, B, Yu, Z, Chen, A, Geiger, A, Gao, S. 2d gaussian splatting for geometrically accurate radiance fields. In: ACM SIGGRAPH 2024 conference papers. 2024, p. 1–11.

[10] Svitov, D, Morerio, P, Agapito, L, Del Bue, A. Billboard splatting (bbsplat): Learnable textured primitives for novel view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2025, p. 25029–25039.

[11] Livny, Y, Pirk, S, Cheng, Z, Yan, F, Deussen, O, Cohen-Or, D, et al. Texture-lobes for tree modelling. ACM Trans Graph 2011;30(4). doi:`10.1145/2010324.1964948`.

[12] Ester, M, Kriegel, HP, Sander, J, Xu, X, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd; vol. 96. 1996, p. 226–231.

[13] Prusinkiewicz, P, Lindenmayer, A. The algorithmic beauty of plants. Springer Science & Business Media; 2012,.

[14] Manfredi, G, Capece, N, Erra, U, Gruosso, M. Treesketchnet: From

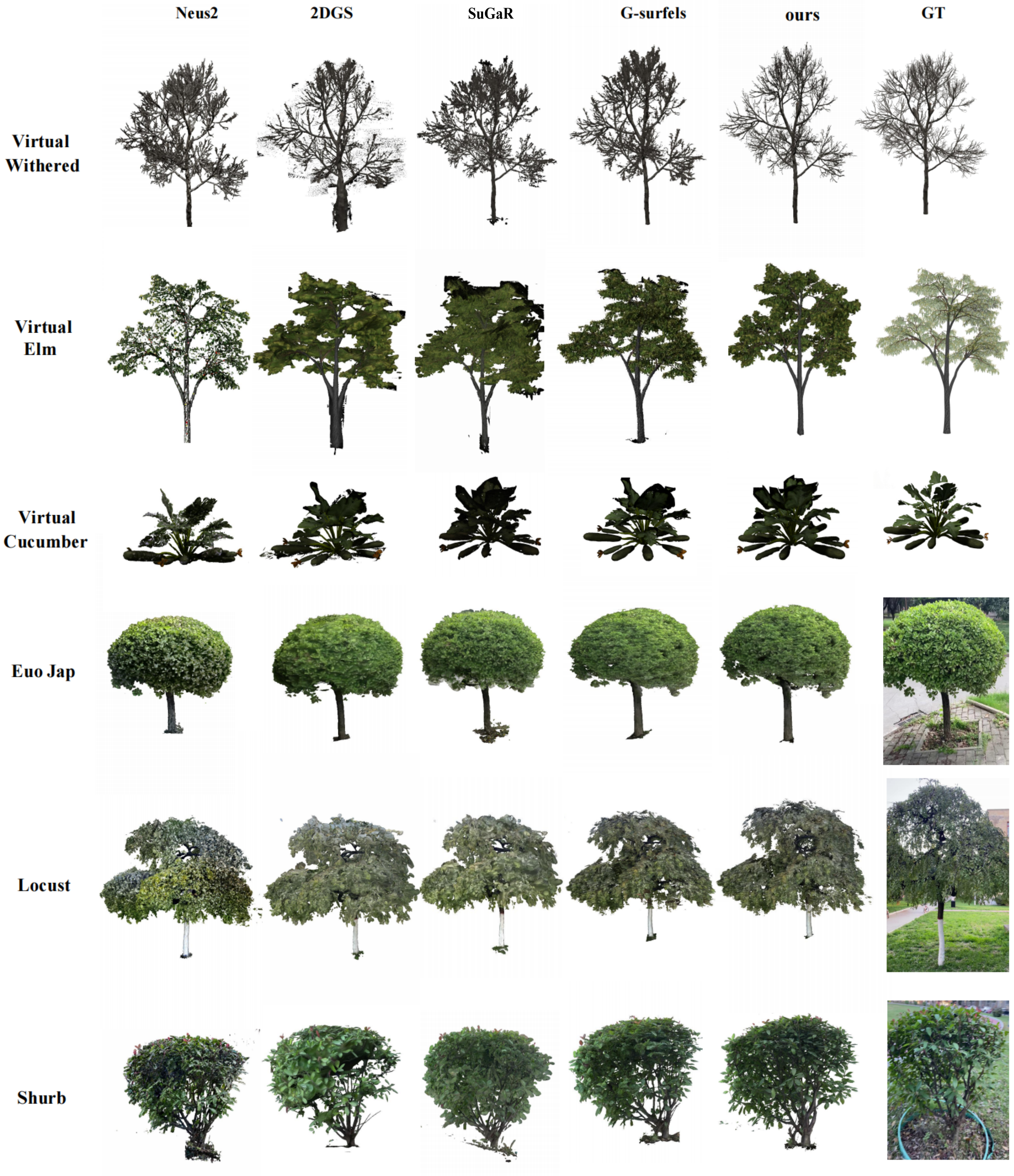

Fig. 11. Comparison of reconstruction quality for six typical vegetation types.

sketch to 3d tree parameters generation. ACM Transactions on Intelligent Systems and Technology 2023;14(3):1–29.

[15] Runions, A, Lane, B, Prusinkiewicz, P. Modeling trees with a space colonization algorithm. In: Proceedings of the Third Eurographics Conference on Natural Phenomena. NPH'07; Goslar, DEU: Eurographics Association. ISBN 9783905673494; 2007, p. 6370.

[16] Xu, H, Gossett, N, Chen, B. Knowledge and heuristic-based modeling of laser-scanned trees. ACM Trans Graph 2007;26(4):19es. doi:10.1145/1289603.1289610.

[17] Livny, Y, Yan, F, Olson, M, Chen, B, Zhang, H, El-Sana, J. Automatic reconstruction of tree skeletal structures from point clouds. In: ACM SIGGRAPH Asia 2010 papers. 2010, p. 1–8.

[18] Wang, Y, Chang, X, Ning, X, Zhang, J, Shi, Z, Zhao, M, et al. Tree branching reconstruction from unilateral point clouds. In: Transactions on Edutainment VIII. Springer; 2012, p. 250–263.

[19] Du, S, Lindenbergh, R, Ledoux, H, Stoter, J, Nan, L. Adtree: Accurate, detailed, and automatic modelling of laser-scanned trees. Remote Sensing 2019;11(18):2074.

[20] Dobbs, H, Batchelor, O, Green, R, Atlas, J. Smart-tree: Neural medial axis approximation of point clouds for 3d tree skeletonization. In: Iberian Conference on Pattern Recognition and Image Analysis(IbPRIA). Springer; 2023, p. 351–362.

[21] Reche-Martinez, A, Martin, I, Drettakis, G. Volumetric reconstruction and interactive rendering of trees from photographs. ACM Transactions on Graphics(TOG) 2004;23(3):720727. doi:10.1145/1015706.1015785.

[22] Neubert, B, Franken, T, Deussen, O. Approximate image-based tree-modeling using particle flows. In: ACM SIGGRAPH 2007 Papers. SIGGRAPH '07; New York, NY, USA. ISBN 9781450378369; 2007, p. 88es. doi:10.1145/1275808.1276487.

[23] Tan, P, Zeng, G, Wang, J, Kang, SB, Quan, L. Image-based tree modeling. ACM Transactions on Graphics(TOG) 2007;26(3):87es. doi:10.1145/1276377.1276486.

[24] Teng, CH, Chen, YS. Image-based tree modeling from a few images with very narrow viewing range. The Visual Computer 2009;25:297–307.

[25] Guo, J, Xu, S, Yan, DM, Cheng, Z, Jaeger, M, Zhang, X. Realistic procedural plant modeling from multiple view images. IEEE Transactions on Visualization and Computer Graphics 2020;26(2):1372–1384. doi:10.1109/TVCG.2018.2869784.

[26] Li, B, Kałużny, J, Klein, J, Michels, DL, Pałubicki, W, Benes, B, et al. Learning to reconstruct botanical trees from single images. ACM Transactions on Graphics (TOG) 2021;40(6):1–15.

[27] Yu, A, Ye, V, Tancik, M, Kanazawa, A. pixelnerf: Neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR). 2021, p. 4578–4587.

[28] Barron, JT, Mildenhall, B, Verbin, D, Srinivasan, PP, Hedman, P. Zip-nerf: Anti-aliased grid-based neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision(ICCV). 2023, p. 19697–19705.

[29] Barron, JT, Mildenhall, B, Tancik, M, Hedman, P, Martin-Brualla, R, Srinivasan, PP. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF international conference on computer vision. 2021, p. 5855–5864.

[30] Reiser, C, Szeliski, R, Verbin, D, Srinivasan, P, Mildenhall, B, Geiger, A, et al. Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. ACM Transactions on Graphics (TOG) 2023;42(4):1–12.

[31] Liu, L, Gu, J, Zaw Lin, K, Chua, TS, Theobalt, C. Neural sparse voxel fields. Advances in Neural Information Processing Systems(NeurIPS) 2020;33:15651–15663.

[32] Hedman, P, Srinivasan, PP, Mildenhall, B, Barron, JT, Debevec, P. Baking neural radiance fields for real-time view synthesis. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV). 2021, p. 5875–5884.

[33] Müller, T, Evans, A, Schied, C, Keller, A. Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (TOG) 2022;41(4):102:1–102:15. doi:10.1145/3528223.3530127.

[34] Yan, Z, Low, WF, Chen, Y, Lee, GH. Multi-scale 3d gaussian splatting for anti-aliased rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, p. 20923–20931.

[35] Yu, Z, Chen, A, Huang, B, Sattler, T, Geiger, A. Mip-splatting: Alias-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2024, p. 19447–19456.

[36] Fan, Z, Wang, K, Wen, K, Zhu, Z, Xu, D, Wang, Z. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. Advances in neural information processing systems 2024;37:140138–140158.

[37] Lee, JC, Rho, D, Sun, X, Ko, JH, Park, E. Compact 3d gaussian representation for radiance field. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, p. 21719–21728.

[38] Papantonakis, P, Kopanas, G, Kerbl, B, Lanvin, A, Drettakis, G. Reducing the memory footprint of 3d gaussian splatting. Proceedings of the ACM on Computer Graphics and Interactive Techniques 2024;7(1):1–17.

[39] Park, JJ, Florence, P, Straub, J, Newcombe, R, Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition(CVPR). 2019, p. 165–174.

[40] Atzmon, M, Haim, N, Yariv, L, Israelov, O, Maron, H, Lipman, Y. Controlling neural level sets. In: Advances in Neural Information Processing Systems; vol. 32. 2019,.

[41] Mescheder, L, Oechsle, M, Niemeyer, M, Nowozin, S, Geiger, A. Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition(CVPR). 2019, p. 4460–4470.

[42] Wang, P, Liu, L, Liu, Y, Theobalt, C, Komura, T, Wang, W. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. NeurIPS 2021 2021;.

[43] Fu, Q, Xu, Q, Ong, YS, Tao, W. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. Advances in Neural Information Processing Systems (NeurIPS) 2022;35:3403–3416.

[44] Darmon, F, Bascle, B, Devaux, JC, Monasse, P, Aubry, M. Improving neural implicit surfaces geometry with patch warping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). 2022, p. 6260–6269.

[45] Wang, Y, Han, Q, Habermann, M, Daniilidis, K, Theobalt, C, Liu, L. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2023, p. 3295–3306.

[46] Li, Z, Müller, T, Evans, A, Taylor, RH, Unberath, M, Liu, MY, et al. Neuralangelo: High-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (ICCV). 2023, p. 8456–8465.

[47] Guédon, A, Lepetit, V. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2024, p. 5354–5363.

[48] Lacewell, D, Edwards, D, Shirley, P, Thompson, W. Stochastic billboard clouds for interactive foliage rendering. Journal of Graphics Tools 2006;11:1–12. doi:10.1080/2151237X.2006.10129213.

[49] Kirillov, A, Mintun, E, Ravi, N, Mao, H, Rolland, C, Gustafson, L, et al. Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. 2023, p. 4015–4026.

[50] Dai, P, Xu, J, Xie, W, Liu, X, Wang, H, Xu, W. High-quality surface reconstruction using gaussian surfels. In: ACM SIGGRAPH 2024 Conference Papers. 2024, p. 1–11.

[51] Kazhdan, M, Hoppe, H. Screened poisson surface reconstruction. ACM Transactions on Graphics (ToG) 2013;32(3):1–13.