HAME-NeRF: High Accuracy Mesh Extraction Leveraging Neural Radiance Fields

Panagiotis Frasiolas[1], Grigogios Aris Cheimariotis[1], Panos K. Papadopoulos[1], and Dimitrios Zarpalas[1]

Information Technologies Institute (ITI), Centre for Research and Technology Hellas (CERTH)

Abstract. Neural Radiance Fields (NeRFs) have transformed image-based 3D reconstruction through differentiable volumetric rendering, enabling high-quality novel view synthesis. However, their implicit volumetric nature is incompatible with the polygonal meshes needed for real-time graphics and simulation applications. The proposed model defines the volume density function as the Secant Hyperbolic Function applied to a signed distance function (SDF) representation. To enable accurate surface representation, the sharpness of the density transition is modulated by a spatially-varying parameter $\beta(x)$, which is learned through a multilayer perceptron (MLP). Experimental results on the NeRF-Synthetic and Mip-NeRF 360 datasets demonstrate improved surface reconstruction accuracy and visual quality compared to NeRF2Mesh, highlighting the effectiveness of the proposed enhancements for efficient and high-fidelity real-time scene reconstruction.

Keywords: NERF · SDF · Mesh

1 Introduction

Reconstruction of high-fidelity 3D surfaces from 2D images is a fundamental problem in computer vision, with applications spanning robotics, photogrammetry, and AR/VR. Many of these applications depend on precise 3D geometry to enable physics-based simulations, real-time visualization, and seamless integration into graphics pipelines. Recent advances in Neural Radiance Fields (NeRF) [6] have significantly improved novel view synthesis (NVS), allowing for photo-realistic rendering of complex scenes. However, while NeRF excels in generating high-quality renderings, it is not inherently optimized for accurate 3D geometry extraction.

The volumetric representation in NeRF enables the synthesis of novel views through differentiable volumetric rendering, but the implicit nature of its scene encoding poses challenges for downstream applications. Unlike conventional polygonal meshes, which are widely used in 3D graphics due to their hardware efficiency and ease of manipulation, NeRF relies on ray marching and implicit functions that lack direct compatibility with standard rendering engines. Moreover, the underlying geometry in NeRF is not explicitly defined as a level-set surface,

often resulting in approximations through dense volumetric regions rather than well-defined object boundaries. This makes extracting accurate and editable 3D meshes from NeRF particularly difficult [2], [3].

This paper presents a framework for extracting high-fidelity, textured surface meshes from RGB images. First, a grid-based NeRF is trained to decompose appearance into diffuse and specular components and extract a coarse geometry via a density field. Then, the initial mesh is refined through differentiable rendering and an iterative mesh refinement algorithm, adjusting vertex positions and face density based on reprojected 2D rendering errors. The proposed implementation incorporates SDF-based density modeling. To the best of our knowledge, this method is the first to optimize the SDF parameter that controls the sharpness of the surface boundary, commonly referred to as β (beta) parameter. Thus, this parameter becomes learned and spatially-varying, enabling sharper surface representation and better geometric control.

The proposed method was evaluated on two benchmark datasets: NeRF-Synthetic [6] and Mip-NeRF 360 [1]. For the NeRF-Synthetic dataset, which provides access to ground truth meshes, geometric accuracy is evaluated using Chamfer Distance and Normal Consistency metrics. These measures specifically assess the quality of the reconstructed shape geometry, capturing both spatial alignment and surface normal fidelity between the predicted and reference meshes. Peak Signal-to-Noise Ratio (PSNR) was used to assess rendering quality on both datasets. Additionally, a qualitative analysis was conducted across all scenes. Experimental results show that the proposed method outperforms NeRF2Mesh [9] in both Chamfer Distance, Normal Consistency, PSNR and visual quality on the NeRF-Synthetic dataset, and achieves comparable, though slightly lower, performance on the more challenging Mip-NeRF 360 dataset.

2 Related Work

2.1 NeRF and Efficient Scene Reconstruction

Neural Radiance Fields (NeRF) have significantly advanced 3D scene reconstruction by representing scenes as continuous volumetric fields parameterized by neural networks. While achieving impressive photorealistic rendering, the original NeRF formulation suffers from slow optimization and difficulties handling unbounded scenes. Subsequent methods, such as Instant-NGP [7] and SNeRG [4], introduce explicit 3D structures like multi-resolution hash grids and sparse voxel grids to accelerate training and enable real-time rendering. However, these approaches often compromise geometric fidelity, limiting their utility for high-precision reconstruction tasks.

2.2 Surface Reconstruction from Implicit Fields

An alternative to volumetric rendering is surface-based representation using Signed Distance Functions (SDFs), which enable high-quality mesh extraction

through algorithms like Marching Cubes [5]. Techniques such as the NeuS [10] model SDFs with differentiable rendering frameworks, improving surface accuracy while maintaining learnability from 2D supervision. To further address challenges with thin structures and complex topologies, hybrid methods such as BakedSDF [11] and NVdiffrec [8] combine volumetric and surface-based representations, optimizing both geometry and appearance for real-time or high-fidelity mesh extraction.

2.3 Bridging Implicit and Explicit Representations

Transforming implicit volumetric models into explicit, textured surface meshes remains an active area of research. Approaches like NeRF2Mesh aim to bridge this gap by extracting surfaces directly from trained NeRFs while preserving photorealistic details. Recent methods increasingly leverage differentiable rendering pipelines to refine surface geometry with 2D supervision, improving mesh quality without sacrificing appearance. Despite these advances, achieving real-time, high-fidelity reconstruction that balances photorealism, geometric precision, and computational efficiency continues to be a core challenge, motivating further exploration of hybrid representations and optimization strategies.

3 Method

The proposed framework extracts a textured surface mesh from a set of RGB images, extending the NeRF2Mesh pipeline. The process comprises two main stages (Fig. 1): an initialization stage using a grid-based NeRF to obtain coarse geometry and appearance, followed by a refinement stage improving both accuracy and visual quality.

Unlike the baseline, which applies Marching Cubes directly to a density field, the proposed method introduces a Signed Distance Function (SDF) for mesh extraction. This provides a more accurate geometric proxy and better surface localization.

The SDF incorporates two key innovations: (a) volumetric density modeling through a hyperbolic secant function for sharper boundaries, and (b) spatially varying sharpness control, with the parameter β learned via an auxiliary MLP across (x, y, z) coordinates.

3.1 Initialization of Geometry and Appearance

A volumetric NeRF structure with an SDF component serves as the foundation for initializing geometry and appearance.

Geometry Initialization Geometry is modeled with a density grid and a shallow MLP:

$$\text{SDF} = \text{MLP}(E^{\text{geo}}(\mathbf{x})), \tag{1}$$

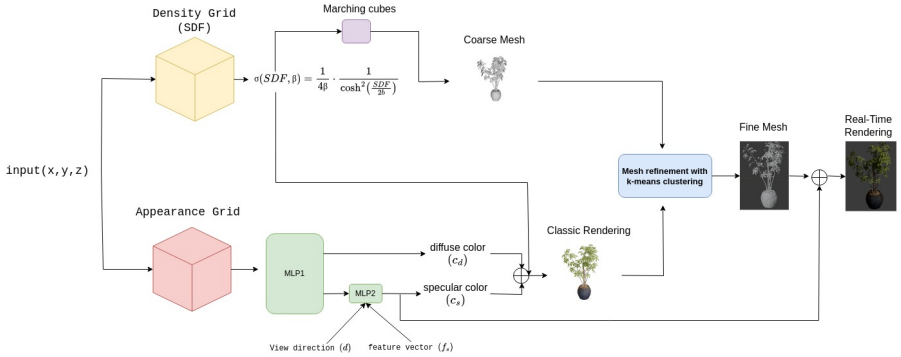


Fig. 1: Overview of the proposed method. Geometry is produced by a density grid, and view-dependent appearance is handled by a separate appearance grid.

where E^{geo} is a multi-resolution feature grid and $\mathbf{x} \in \mathbb{R}^3$ denotes spatial coordinates.

The density function $\sigma(SDF, \beta)$ is designed to concentrate density near the zero-level set using a hyperbolic cosine:

$$\sigma(SDF, \beta) = \frac{1}{4\beta} \cdot \frac{1}{\cosh^2\left(\frac{SDF}{2\beta}\right)}. \tag{2}$$

$$\beta(x) = \text{MLP}_\beta(x) \tag{3}$$

Unlike prior methods that employ a fixed β value, the proposed approach learns a spatially varying $\beta(\mathbf{x})$ through an MLP, enhancing flexibility for complex surfaces. The ablation study compares the model's performance using a fixed beta value versus a spatially varying beta.

Appearance Initialization Appearance is decomposed into view-independent diffuse color $\mathbf{c}_d$ and view-dependent specular color $\mathbf{c}_s$:

$$\mathbf{c}_d, \mathbf{f}_s = \psi(\text{MLP}_1(E^{\text{app}}(\mathbf{x}))), \quad \mathbf{c}_s = \psi(\text{MLP}_2(\mathbf{f}_s, \mathbf{d})), \tag{4}$$

where $\mathbf{f}_s$ encodes specular information, $\mathbf{d}$ is the view direction, and ψ denotes a sigmoid activation. The final color is:

$$\mathbf{c} = \mathbf{c}_d + \mathbf{c}_s. \tag{5}$$

This separation allows for efficient baking of diffuse textures and flexible handling of specularities using lightweight shaders, following [3]. As in NeRF2Mesh, lighting is pre-baked into textures to avoid complex light estimation.

3.2 Mesh Refinement and Optimization

After initial convergence, a coarse mesh is extracted via Marching Cubes. This mesh undergoes refinement to improve both geometry and appearance.

Differentiable rendering with nvdiffrast is employed to align the projected mesh with input images. The appearance model is reused to accelerate convergence. Vertex positions are optimized by learning per-vertex offsets, using gradients from a pixel-wise rendering loss.

To adapt mesh resolution, face refinement is performed heuristically: faces with high rendering errors are identified using k-means clustering. High-error faces are subdivided, while low-error regions are simplified, concentrating detail where needed. This adaptive process repeats during training.

For unbounded scenes, the environment is partitioned into regions of progressively lower resolution away from the center. Regularization terms are applied to ensure smooth surfaces and bounded vertex displacements.

3.3 Loss Function

Training minimizes a weighted combination of three objectives:

Photometric Rendering Loss The rendering loss encourages consistency between rendered and ground-truth pixel colors:

$$\mathcal{L}_{\text{render}} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\mathbf{c}}_i - \mathbf{c}_i\|_2^2 . \tag{6}$$

Eikonal Loss The Eikonal loss regularizes the SDF to behave like a true distance field:

$$\mathcal{L}_{\text{eik}} = \left(\|\nabla_{\mathbf{x}} \text{SDF}(\mathbf{x})\|_2 - 1\right)^2 . \tag{7}$$

Beta Regularization Loss An L_1 penalty on $\beta(\mathbf{x})$ prevents unstable variations and encourages smooth transitions:

$$\mathcal{L}_{\beta} = \lambda_{\beta} \cdot |\beta(\mathbf{x})| . \tag{8}$$

Total Loss The total training objective is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{render}} + \lambda_{\text{eik}} \mathcal{L}_{\text{eik}} + \lambda_{\beta} \mathcal{L}_{\beta}. \tag{9}$$

3.4 Evaluation

Datasets: The effectiveness and generalization capability of the proposed method are evaluated using two diverse datasets:

1. NeRF-Synthetic dataset [6]: A widely used benchmark introduced in the original NeRF, containing various synthetic scenes. Each scene has RGB images with known camera poses, ideal for testing novel view synthesis.
2. Mip-NeRF 360 dataset [1]: A collection of nine complex scenes (five outdoor and four indoor), each featuring a detailed central object or area surrounded by intricate backgrounds.

Fig. 2: Final output comparison on Nerf-Synthetic Dataset. The output is the combination of the fine mesh with the small MLP for illuminations.

Metrics: To evaluate geometric similarity between two 3D meshes, the Chamfer Distance (CD) is computed between surface points from each mesh. This metric quantifies the average distance between points on one surface and their nearest neighbors on the other, offering a robust measure of alignment and shape similarity. An observability grid of resolution 256 is constructed using rays cast from test views located within the mesh bounding box.

Normal Consistency measures how well the surface normals of a predicted mesh align with those of a reference mesh. For each predicted point, the closest reference point is found, and the cosine similarity between their normals is computed. Flipped normals are corrected before averaging. This metric captures the local surface orientation accuracy, which is critical for realistic shading, ren-

dering, and simulation. Higher values indicate better geometric fidelity on a fine scale.

Peak Signal-to-Noise Ratio (PSNR) is used to evaluate the photometric accuracy of the rendered images by comparing them to ground-truth images.

$$\text{PSNR} = 20 \cdot \log_{10} \left(\frac{\text{MAX}}{\text{MSE}} \right) \tag{10}$$

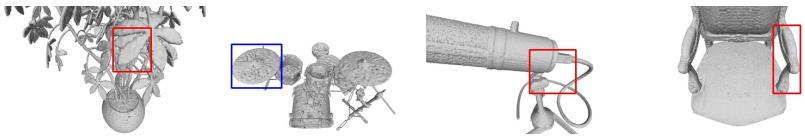
NeRF-Synthetic dataset: Qualitative comparison on the NeRF-Synthetic dataset is presented to assess the quality of the coarse mesh extracted during the initial stage of the pipeline (see Fig. 3), the fine mesh extracted in the last stage (see Fig. 4) and the final rendered image (see Fig. 2)

Figure 3 shows the coarse mesh extracted during the initial stage of the pipeline. The figure includes images from four different scenes in the dataset. Red squares highlight areas where fine geometric details are more accurately captured and the surface appears smoother. Blue squares indicate regions where the mesh surface is more complete and better filled, with fewer missing parts.

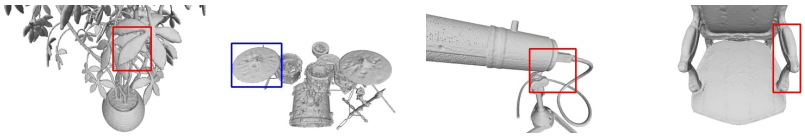
Figure 4 shows the fine mesh extracted in the final stage. Zoomed-in views are also provided, with red squares highlighting regions that exhibit smoother surfaces or improved mesh topology, characterized by fewer and more uniform faces.

Figure 4 demonstrates that the rendered output of the proposed method closely matches the ground truth, with minimal visible differences.

Across all scenes, the proposed method outperforms NeRF2Mesh by achieving lower Chamfer Distance (see Table 1), higher Normal Consistency (see Table 2), and higher PSNR (see Table 3). These results confirm its superiority in both geometric reconstruction and rendering quality.

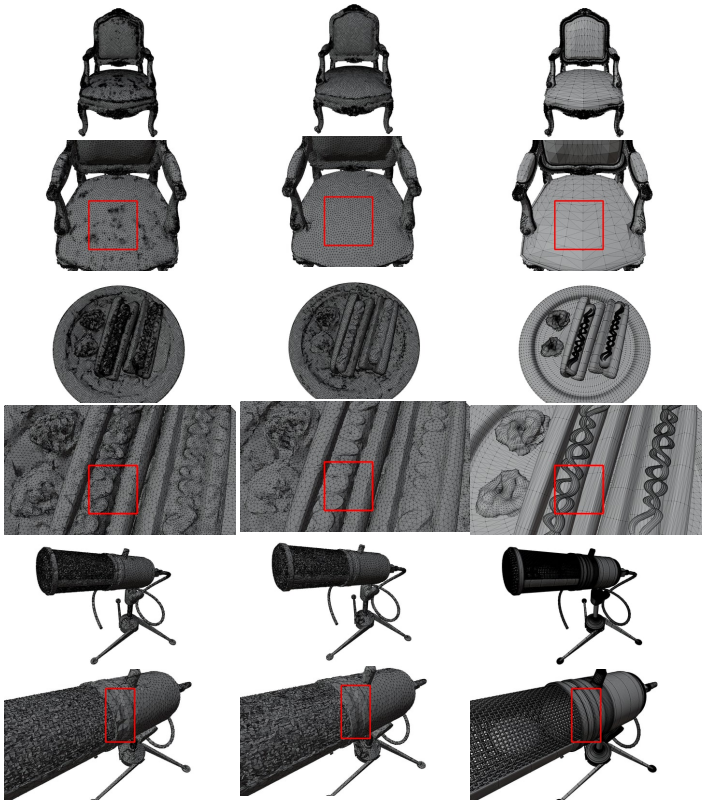


(a) Nerf2Mesh



(b) Proposed Method

Fig. 3: Comparison of generated coarse meshes on Nerf-Synthetic Dataset.



(a) Nerf2 Mesh (b) Proposed Method (c) Ground Truth

Fig. 4: Geometry comparison of the generated fine mesh on Nerf-Synthetic Dataset.

Table 1: Chamfer Distance (Unit is 10^{-3}) across NeRF synthetic dataset scenes. Bold numbers indicate best results per scene.

	Chair	Drums	Ficus	Hotdog	Materials	Mic	Ship
Nerf2Mesh [9]	**7.677**	0.6010	0.0439	70.68	**0.407**	0.2060	276.6
Proposed Method	7.825	**0.5761**	**0.0232**	**9.151**	0.468	**0.0680**	**51.56**

Table 2: Normal Consistency across NeRF synthetic dataset scenes. Bold numbers indicate best results per scene.

	Chair	Drums	Ficus	Hotdog	Materials	Mic	Ship
Nerf2Mesh [9]	**0.9077**	0.8316	0.8775	0.8243	**0.8781**	0.8636	0.6346
Proposed Method	0.9064	**0.8519**	**0.8819**	**0.8439**	0.8344	**0.8744**	**0.6387**

Table 3: PSNR across NeRF synthetic dataset scenes. Bold numbers indicate best results per scene.

	Chair	Drums	Ficus	Hotdog	Materials	Mic	Ship
Nerf2Mesh [9]	32.3917	25.0147	30.0653	**34.9732**	25.7856	31.9232	27.4562
Proposed Method	**32.4012**	**25.0215**	**30.4823**	34.9219	**25.9793**	**31.9340**	**27.5306**

Mip-Nerf 360 dataset: A qualitative comparison on the Mip-NeRF 360 dataset is presented to assess the quality of the fine mesh extracted in the final stage (see Fig.5) and the corresponding rendered images (see Fig.6). Red squares indicate regions where the proposed method yields cleaner geometry, whereas Nerf2Mesh tends to generate a high number of mesh faces without significant geometric improvement. However, a limitation of the proposed method is its slightly lower performance in terms of PSNR (see Table 4)

Figure 6 demonstrates that the rendered output of the proposed method closely matches the ground truth, with minimal visible differences.

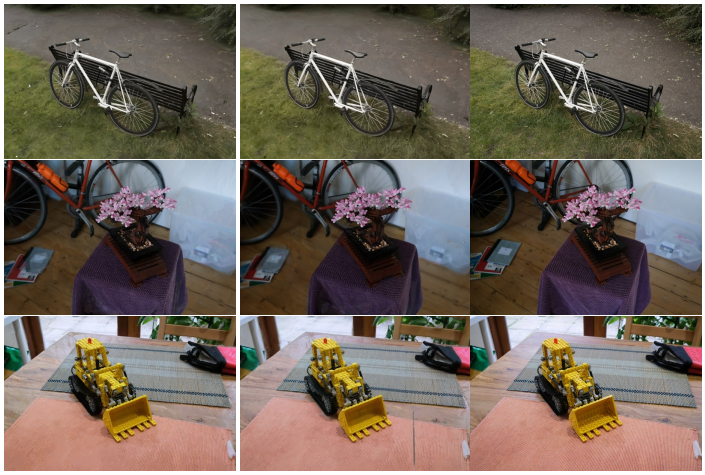
Table 4: PSNR across Mip-NeRF 360 dataset scenes. Bold numbers indicate best results per scene.

	Bicycle	Bonsai	Counter	Garden	Kitchen	Room	Stump
Nerf2Mesh [9]	**22.4764**	**26.1323**	**24.8213**	**22.6985**	**25.7852**	**26.882**	**23.1960**
Proposed Method	22.1009	25.1960	24.6931	22.6091	24.6916	24.3575	23.0895

(a) Nerf2Mesh

(b) Proposed Method

Fig. 5: Geometry comparison of the generated fine mesh Mip-NeRF 360 scenes.



(a) Nerf2Mesh (b) Proposed Method (c) Ground Truth

Fig. 6: Final output comparison on Mip-Nerf 360. The output is the combination of the fine mesh with the small MLP for illuminations.

4 Conclusion

This paper presents a robust NeRF-based framework for extracting high-quality textured surface meshes using a signed distance function with spatially adaptive sharpness. A hyperbolic secant-based density formulation enables sharper surface localization and enhances mesh fidelity, particularly in areas with fine structural detail. The process begins with geometry and appearance initialized through a volumetric NeRF, followed by mesh refinement using differentiable rendering. While effective, the framework inherits common limitations of mesh-based pipelines, including baked-in lighting that hampers relighting capabilities and reduced accuracy in capturing complex view-dependent effects. Future work may focus on improving appearance modeling and adopting more expressive rendering techniques to further enhance realism and flexibility.

Acknowledgments. This work was supported by the European Union's Horizon Europe programme under grant number 101092875 "DIDYMOS-XR" (https://www.didymos-xr.eu). The computational resources were granted with the support of GRNET.

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022)

2. Chen, W., Ling, H., Gao, J., Smith, E., Lehtinen, J., Jacobson, A., Fidler, S.: Learning to predict 3d objects with an interpolation-based differentiable renderer. Advances in neural information processing systems **32** (2019)
3. Chen, Z., Funkhouser, T., Hedman, P., Tagliasacchi, A.: Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16569–16578 (2023)
4. Hedman, P., Srinivasan, P.P., Mildenhall, B., Barron, J.T., Debevec, P.: Baking neural radiance fields for real-time view synthesis. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5875–5884 (2021)
5. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Seminal graphics: pioneering efforts that shaped the field, pp. 347–353 (1998)
6. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**, 99–106 (2021)
7. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG) **41**(4), 1–15 (2022)
8. Munkberg, J., Hasselgren, J., Shen, T., Gao, J., Chen, W., Evans, A., Müller, T., Fidler, S.: Extracting triangular 3d models, materials, and lighting from images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8280–8290 (2022)
9. Tang, J., Zhou, H., Chen, X., Hu, T., Ding, E., Wang, J., Zeng, G.: Delicate textured mesh recovery from nerf via adaptive surface refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17739–17749 (2023)
10. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689 (2021)
11. Yariv, L., Hedman, P., Reiser, C., Verbin, D., Srinivasan, P.P., Szeliski, R., Barron, J.T., Mildenhall, B.: Bakedsdf: Meshing neural sdfs for real-time view synthesis. In: ACM SIGGRAPH 2023 Conference Proceedings. pp. 1–9 (2023)