Inverse Rendering for High-Genus Surface Meshes from Multi-View Images

Xiang Gao
Futurewei Technologies
Stony Brook University
gao2@cs.stonybrook.edu

Xinmu Wang
Futurewei Technologies
Stony Brook University
xinmuwang@cs.stonybrook.edu

Xiaolong Wu
Futurewei Technologies
Purdue University
wu1565@purdue.edu

Jiazhi Li
Futurewei Technologies
University of Southern California
jiazhil@usc.edu

Jingyu Shi
Futurewei Technologies
Purdue University
shi537@purdue.edu

Yuanpeng Liu
Futurewei Technologies
Stony Brook University
yuanpliu@cs.stonybrook.edu

Yu Guo
Futurewei Technologies
yguo1@futurewei.com

Xiyun Song
Futurewei Technologies
xsong@futurewei.com

Heather Yu
Futurewei Technologies
hyu@futurewei.com

Zongfang Lin
Futurewei Technologies
zlin1@futurewei.com

Xianfeng David Gu
Stony Brook University
gu@cs.stonybrook.edu

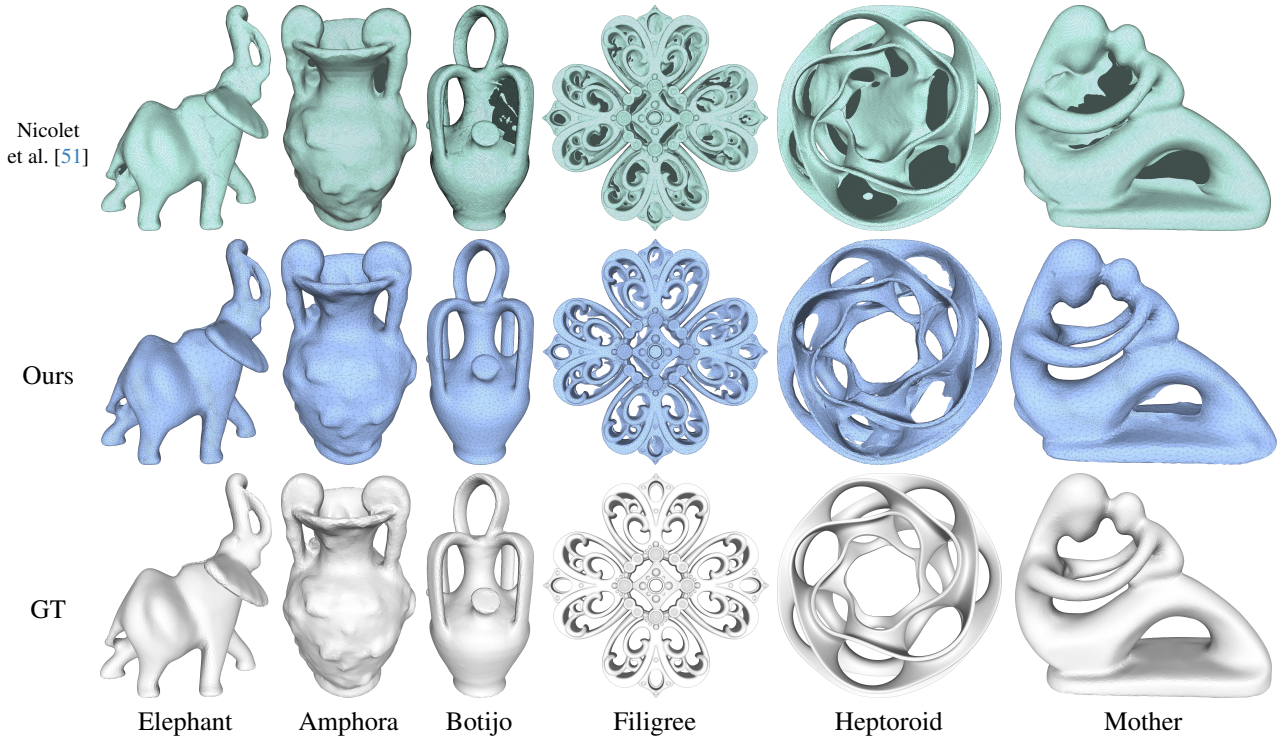

Figure 1. **Inverse Rendering for High-Genus Surface Meshes from Multi-View Images.** (Top) Reconstructions using the **SOTA** method [51], which **produces incorrect genus number, leading to incorrect topology**. (Middle) Our method with the **correct genus number, leading to correct topology**. (Bottom) Challenging High-Genus Ground Truth. Please see **Appendix D.3** for quantitative results.

Abstract

We present a topology-informed inverse rendering approach for reconstructing high-genus surface meshes from multi-view images. Compared to 3D representations like voxels and point clouds, mesh-based representations are preferred as they enable the application of differential geometry theory and are optimized for modern graphics pipelines. However, existing inverse rendering methods often fail catastrophically on high-genus surfaces, leading to the loss of key topological features, and tend to over-smooth low-genus surfaces, resulting in the loss of surface details. This failure stems from their overreliance on Adam-based optimizers, which can lead to vanishing and exploding gradients. To overcome these challenges, we introduce an adaptive V-cycle remeshing scheme in conjunction with a re-parametrized Adam optimizer to enhance topological and geometric awareness. By periodically coarsening and refining the deforming mesh, our method informs mesh vertices of their current topology and geometry before optimization, mitigating gradient issues while preserving essential topological features. Additionally, we enforce topological consistency by constructing topological primitives with genus numbers that match those of ground truth using Gauss-Bonnet theorem. Experimental results demonstrate that our inverse rendering approach outperforms the current state-of-the-art method, achieving significant improvements in Chamfer Distance and Volume IoU, particularly for high-genus surfaces, while also enhancing surface details for low-genus surfaces.

1. Introduction

Inverse reconstruction of high-quality 3D surfaces from images is one of the most challenging tasks in computer vision and computer graphics, with diverse applications spanning virtual and augmented reality (VR/AR), medical imaging, robotics, autonomous driving, and 3D printing. Recent advances in reconstruction [13, 17, 22, 28, 46, 66, 70], generative modeling [33–36, 56, 64, 68], and inverse rendering [12, 26, 44, 45, 51, 52] have led to significant improvements in geometric precision and visual realism, representing a major leap toward real-world deployment of images-driven 3D geometry creation. However, choosing the right 3D representation has remained a critical challenge due to the inherent trade-offs in each 3D representation. For example, voxels [14, 43, 67, 71] offer structured 3D grids but are computationally expensive at high resolutions. Point clouds [15, 41, 50, 74] capture surface points efficiently but lack connectivity, complicating the detection of small holes and other topological features. Triangular and tetrahedral meshes [17, 18, 20, 26, 51, 52] provide detailed connected surface representations capturing fine geometric details but rely on complex data structures for efficient local mesh processing [4, 9, 16] due to its irregular structure. Signed Distance Fields (SDFs) [19, 25, 44, 45, 54, 65, 69] enable smooth and continuous surface representation, but incur high computational cost due to the need for dense 3D grid evaluation and frequent distance queries, and often suffer from loss of surface details during surface mesh extraction via Marching Cubes [38].

Among these 3D representations, meshes are particularly favored in physical simulations and graphics rendering, as modern graphics pipelines are highly optimized for processing *mesh-based* structures efficiently. In engineering, mesh representations are widely used to solve partial differential equations (PDEs). *Mesh-based* finite element simulations have been extensively applied in fields such as solid mechanics [32, 62], fluid mechanics [1, 5], aerodynamics [11, 57], electromagnetics [53], geophysics [58, 60], and acoustics [42], due to their ability to capture intricate geometric details essential for accurate analysis. Furthermore, most mathematical proofs regarding convergence, as well as theoretical analyses of consistency, stability, and error bounds, are specifically developed for *mesh-based* discretizations [55]. Utilizing meshes also allows us to apply theories from discrete differential geometry [4, 9, 16] to inspect surface features such as mean, Gaussian and principal curvatures, enabling the classification of local shapes and the determination of the surface *genus*, a measure of topological complexity that quantities the number of holes or tunnel loops on a surface. Ensuring genus consistency is vital, as mismatches can result in the loss of critical topological features and lead to visually inaccurate reconstructions. Collectively, these factors establish meshes as the standard representation in both engineering and rendering applications, where 3D surfaces must meet rigorous requirements for accuracy and precision. While converting from other representations to triangular meshes can introduce loss of geometric or topological detail, conversions from meshes typically retain nearly all original information. This inherent adaptability makes *mesh-based* representations the preferred choice across a wide range of practical applications.

Recently, there has been a growing shift toward solving inverse rendering problems without relying on neural networks, particularly using *mesh-based* surface representations. Unlike neural-implicit methods such as NeRF [46, 48], neural signed distance fields (SDFs)[54, 65], and 2D/3D Gaussian Splatting [22, 28], which are based on the volume rendering equation, *mesh-based* methods [26, 51, 52] adopt differentiable rasterization pipelines to directly optimize surface geometry. This trend has been largely propelled by recent advances in physics-based differentiable rendering (PBDR) [21, 24, 27, 30, 31, 37, 39, 75] and high-performance GPUs, which enable efficient gradient computation over mesh parameters. Although neural network–free

methods [26, 51, 52] are appealing for *real-time* applications, they remain limited in handling complex geometries, particularly high-genus surface meshes. The root cause lies in their heavy reliance on gradient-based optimizers such as Adam [29], which often leads to *vanishing* or exploding gradients. As a result, the geometry becomes overly smoothed, and key topological features are lost. The reconstructed meshes often suffer from *poor* quality, exhibiting artifacts such as topological inconsistency, non-manifold edges, and self-intersections. These issues significantly limit their applicability to downstream tasks such as relighting, physical simulation, and 3D printing. Addressing these limitations is essential for accurately recovering high-quality, high-genus surface from multi-view images.

To overcome these challenges, we introduce a topology-informed inverse rendering method that leverages an adaptive V-cycle remeshing scheme rooted in geometry processing, in conjunction with an Adam-based optimizer specifically designed for reconstructing high-genus surfaces from multi-view images. By periodically coarsening and refining the mesh and subsequently optimizing, our method effectively mitigates gradient issues and preserves key geometric and topological features. We further enforce topological consistency by constructing topological primitives with genus numbers matching those of the ground truth, utilizing the Gauss-Bonnet theorem from differential geometry. Experimental results demonstrate that our approach surpasses previous methods in terms of reconstructed surface quality, Chamfer Distance, and Volume IoU, particularly for high-genus surface mesh reconstructions.

To summarize, we offer **three principal contributions**:

- We directly address the core challenges of reconstructing high-genus surface meshes using a *mesh-based* representation within an inverse rendering framework.
- We leverage an adaptive V-cycle remeshing scheme in conjunction with an Adam-based optimizer to effectively mitigate gradient issues, enhancing topological awareness, preserving key topological features, and enforcing topological consistency by establishing a homeomorphism between initial and ground truth surfaces, achieving genus invariance.
- We demonstrate superior performance in reconstructing high-genus surfaces, evaluated both qualitatively and quantitatively using Chamfer Distance and Volume IoU.

2. Related Work

3D Reconstruction. Reconstructing 3D objects from multi-view images is inherently an ill-posed problem. As a result, extensive research has been devoted to addressing it. Early neural network-based methods typically rely on 2D image encoders and 3D decoders trained on 3D datasets, using both explicit and implicit representations, including voxels [8, 72, 73] and point clouds [15, 41]. Although promising, these approaches face challenges due to their underlying 3D representations. The former requires high 3D grid resolution to capture fine surface details, leading to significant computational overhead, while the latter relies on dense points to capture topological features such as small holes or tunnels. Furthermore, both voxels and point clouds require post-processing to convert to meshes for applications such as relighting and physical simulation. Recent advancements in implicit representations, particularly neural radiance fields (NeRF) [46], have led to generative models based on neural fields [6, 7], enabling 3D shape learning from 2D images via differentiable rendering. However, converting implicit representations to explicit surfaces often leads to loss of fine geometric detail. Recent methods such as 3D Gaussian Splatting (GS) [28] and TetSphere Splatting [17] have shifted focus back to explicit representations, leveraging Gaussian kernels and tetrahedral meshes, respectively, to improve both visual and reconstruction quality. Tetrahedral meshes enable lossless surface extraction but are more memory-intensive than surface meshes, while 3D Gaussian Splats require post-processing, which may result in loss of surface details. For applications such as relighting, physical simulation, and 3D printing, directly using 3D surface *mesh-based* representation without sacrificing detail is highly desirable.

Inverse Rendering for 3D. Recent advancements in physics-based differentiable rendering (PBDR) [24, 30, 40, 75], largely enabled by high-performance NVIDIA GPUs, have significantly expanded the capabilities of inverse rendering, allowing efficient gradient-based optimization to recover scene parameters such as geometry, texture, and lighting from multi-view images. With Adam-based solvers, recent works [26, 51] have demonstrated the extraction of high-quality 3D triangular meshes in seconds to minutes under consistent lighting, **notably without relying on neural networks**. To handle more complex environmental conditions, approaches such as [19, 47] incorporate neural networks to indirectly generate triangular meshes, textures, and lighting, though these methods incur significantly higher computational cost. In addition, methods like [44, 45] formulate inverse rendering using level set methods to represent surfaces implicitly. These methods are computationally expensive but particularly suited for high-genus topology due to its powerful 3D representation. In contrast, [40] apply inverse rendering to simultaneously recover both geometry and appearance for high-genus surfaces, such as genus-1 shapes. However, reconstructing high-genus surface meshes using *mesh-based* representations without neural networks remains challenging, as most existing methods are not topologically aware. Sole reliance on Adam optimization often produces overly smooth surfaces and suffers from vanishing or exploding gradients, leading to the loss of critical geometric and topological features. Addressing

these limitations is essential to advance *mesh-based* inverse rendering that robustly captures high-genus topology and fine surface detail.

3. Theoretical Foundations

Differential Geometry of Surfaces. Differential geometry provides the mathematical foundation to analyze the curvatures of the surface, which is often represented by a vector-valued function:

$$\vec{\mathbf{r}}(u,v) = (x_1(u,v), x_2(u,v), x_3(u,v)) \in \mathbb{R}^3, \quad (1)$$

where $(u,v) \in \mathbb{R}^2$ are the coordinate parameters of the surface. At each point on the surface, the vectors $\vec{\mathbf{r}}_u = \frac{\partial \vec{\mathbf{r}}}{\partial u}$ and $\vec{\mathbf{r}}_v = \frac{\partial \vec{\mathbf{r}}}{\partial v}$ form a local basis for the tangent plane.

Principal Curvatures. One of the key reasons our adaptive V-Cycle remeshing can be topology-informed and preserves topological and geometric features is that it is guided by principal curvatures k_1 and $k_2 \in \mathbb{R}$, which are defined as:

$$k_1 = \max_{\phi} k_{\vec{\mathbf{n}}}(\phi), \quad k_2 = \min_{\phi} k_{\vec{\mathbf{n}}}(\phi), \quad (2)$$

where $k_{\vec{\mathbf{n}}}(\phi)$ represents the normal curvature in the direction of the polar angle $\phi \in [0, 2\pi)$.

Mean and Gaussian Curvatures. In differential geometry, both the mean curvature and Gaussian curvature at a point on the surface are deeply linked to the principal curvatures. For instance, the mean curvature $H \in \mathbb{R}$ at a point on a surface is defined as the average of the principal curvatures k_1 and k_2:

$$H = \frac{1}{2\pi} \int_0^{2\pi} k_{\vec{\mathbf{n}}}(\phi)\, d\phi = \frac{k_1 + k_2}{2}. \quad (3)$$

Similarly, the Gaussian curvature $K \in \mathbb{R}$ at the same point is defined as the product of the principal curvatures k_1 and k_2:

$$K = k_1 \cdot k_2. \quad (4)$$

Assuming we know both H and K from Equations 3 and 4, we can simply solve for the principal curvatures k_1 and k_2, yielding:

$$k_1, k_2 = H \pm \sqrt{H^2 - K}. \quad (5)$$

In practice, we always triangulate the surface into a mesh, which allows us to compute the discrete mean and Gaussian curvatures, enabling us to approximate the principal curvatures. We will see this in the next few sections.

Discrete Gaussian Curvature. Working on the same mesh, the discrete Gaussian curvature K_i at vertex v_i is given by the angle deficit method [4, 16]:

$$\mathbf{K_i} = \begin{cases} 2\pi - \sum_{j \in N(i)} \theta_j, & \text{if } v_i \notin B, \\ \pi - \sum_{j \in N(i)} \theta_j, & \text{if } v_i \in B, \end{cases} \quad (6)$$

where θ_j are the interior angles at v_i across adjacent triangles, $N(i)$ is the one-ring neighborhood of faces around v_i, and B denotes boundary vertices.

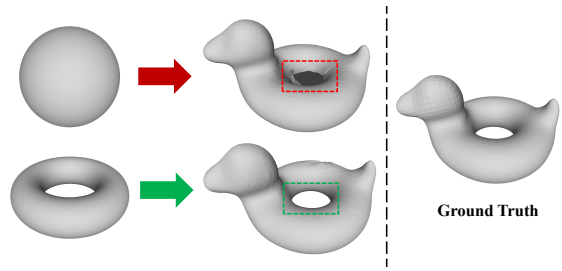

Figure 2. Topological Consistency: The top row shows a sphere with **genus 0**, not homeomorphic to the Bob surface's **genus 1** ground truth, resulting in topological inconsistency. The bottom row shows a torus with **genus 1**, ensuring topological consistency.

Discrete Mean Curvature. Interestingly, by leveraging the discrete Laplacian-Beltrami operator $\mathbf{L}$ [3, 49], we can approximate the mean curvature H_i at each vertex v_i as:

$$H_i = \frac{1}{2} \|\mathbf{L}\vec{\mathbf{x}}_i\|, \quad (7)$$

where $\mathbf{L}\vec{\mathbf{x}}_i = \sum_{j \in N(i)} w_{ij}(\vec{\mathbf{x}}_j - \vec{\mathbf{x}}_i)$ is the discrete Laplacian applied to the position $\vec{\mathbf{x}}_i$ of v_i using either uniform or cotangent weight with the factor $\frac{1}{2}$ included to align with the continuous definition. Hence, we can approximate the principal curvatures using the discrete Gaussian curvature and mean curvatures using earlier Equation 5.

Gauss-Bonnet Theorem. In order to enforce topological consistency as to prevent topological mismatch between initial and ground truth surfaces, we apply the Gauss-Bonnet theorem which links the genus g of a surface S to its Euler characteristic $\chi(S)$ via:

$$\int_S K\, dA + \int_{\partial S} k_g\, ds = 2\pi\chi(S), \quad (8)$$

where K denotes Gaussian curvature, k_g denotes the geodesic curvature along the boundary ∂S, and $\chi(S) = 2 - 2g$ for a closed, orientable surface without boundary. For a triangular mesh $\mathcal{M}$, Euler characteristic is defined as:

$$\chi(S) = |V| + |F| - |E|, \quad (9)$$

where $|V|$, $|E|$, and $|F|$ represent the number of vertices, edges, and faces, respectively. Consequently, the genus g of a closed, orientable surface can be computed as:

$$g = 1 - \frac{|V| + |F| - |E|}{2}. \quad (10)$$

By applying Equations 10, we can compute the genus g of any triangulated 3D surface, enabling us to establish a homeomorphism between initial triangulated surfaces and the ground truth ones to circumvent genus mismatch, as shown in Figure 2. For an in-depth discussion on discrete differential geometry, we recommend [4, 9, 16].

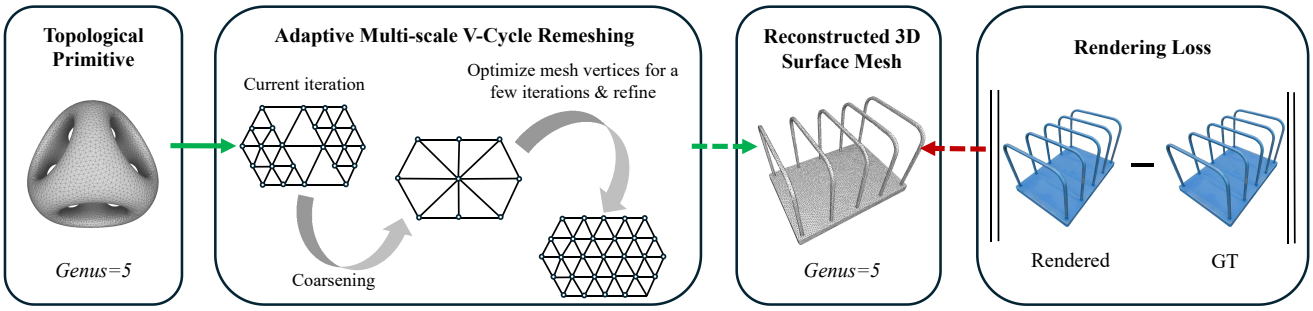

Figure 3. **Overall Pipeline**: A triangulated topological primitive, with genus matching that of the ground truth, undergoes adaptive V-cycle remeshing with periodic coarsening and refining stages, followed by optimization using an Adam-based optimizer to minimize the multiview rendering loss.

4. Method

4.1. Problem Formulation

We formulate the inverse rendering problem for high genus surfaces as:

$$\begin{aligned} \arg\min_{\mathbf{x}} \quad & \Phi(R(\mathbf{x})) + \mathbf{w_1}\,\mathrm{tr}(\mathbf{x}^T\mathbf{L}\mathbf{x}) \\ \text{s.t.} \quad & \det(\mathbf{J}_R^{(k)}) > 0, \quad \forall k \in \{1,\dots,|F|\} \end{aligned} \tag{11}$$

where $\mathbf{x} \in \mathbb{R}^{n\times 3}$ represents the mesh vertex positions, $\mathbf{L} \in \mathbb{R}^{n\times n}$ is the uniform bi-Laplacian matrix [2, 23], $R(\cdot)$ denotes the rendering function, $\Phi(\cdot)$ quantifies the rendering loss between rendered and target images, $\mathrm{tr}(\cdot)$ is the sum of the diagonal entires of a matrix, $|F|$ is the number of faces of the triangular mesh, and $\mathbf{J}_R^{(k)}$ represents the Jacobian for each triangle k. Reformulating the Equation 11 as an unconstrained optimization problem simply yields:

$$\begin{aligned} \arg\min_{\mathbf{x}} & \Phi(R(\mathbf{x})) + \mathbf{w_1}\,\mathrm{tr}(\mathbf{x}^T\mathbf{L}\mathbf{x}) \\ & + \mathbf{w_2}\sum_{k=1}^{|F|}\left(\min\{0, \det(\mathbf{J}_R^{(k)})\}\right)^2 \end{aligned} \tag{12}$$

allowing for optimization via gradient descent solvers. In this formulation, a large $\mathbf{w_1}$ enforces mesh smoothness, while a large $\mathbf{w_2}$ prevents triangle inversion.

4.2. Adaptive V-Cycle Remeshing

Our method leverages adaptive remeshing guided by the principal curvatures k_1 and k_2, derived from Equation 5 and approximated using Equations 6 and 7 . **The overall pipeline** is illustrated in Figure 3. To maintain an optimal mesh structure that balances computational efficiency and topological awareness, periodic coarsening and refining are essential. Coarsening alone reduces computational load by simplifying the mesh but risks losing critical surface details, while refining enhances detail at the cost of increased memory usage. By alternating coarsening and refining, we reposition vertices $\mathbf{x} \in \mathbb{R}^{n\times 3}$ to create a high-quality mesh that accurately captures geometric features while maintaining topological consistency. After each remeshing step, vertex positions are further optimized using a re-parametrized Adam-based optimizer [51] to minimize rendering loss. This iterative process ensures that the surface mesh is informed by topological features for accurate reconstruction of high-genus surfaces while enhancing surface details for low-genus surfaces. **See Appendix C.2 for the adaptive V-Cycle remeshing pseudo-code.**

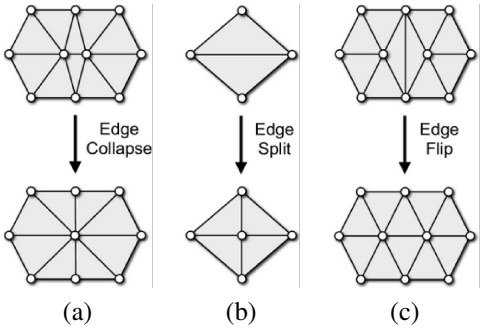

Figure 4. **Visualization of topology-preserving local mesh operations** [61]. (a) Edge collapse, (b) Edge split, and (c) Edge flip.

More specifically, our adaptive remeshing algorithm, leverages the half-edge data structure [3, 4, 16] for coarsening and refinement, proceeds as follows:

1. **Edge Splitting:** We split edges with high curvature (Figure 4a) to enhance resolution in regions with sharp surface details while uniformly splitting low-curvature edges based on the averaged edge length.
2. **Edge Collapsing:** We collapse short edges (Figure 4b) until all edge lengths meet a minimum threshold, optimizing mesh structure by reducing excess vertices where high resolution is unnecessary.
3. **Edge Flipping:** We flip edges (Figure 4c) to improve vertex valence (Targeting a valence of 6 for any closed surface), maintaining mesh stability and uniformity.
4. **Tangential Smoothing:** We smooth vertices in direction parallel to the tangent plane to improve triangle quality.

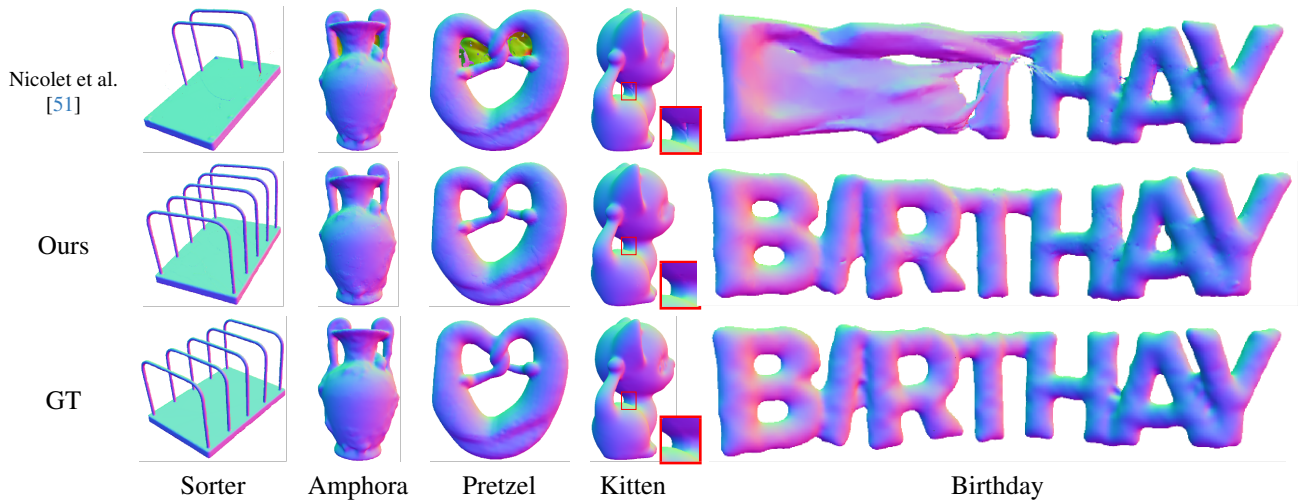

Figure 5. Qualitative High-Genus Reconstruction: Rendered Views in Normal Maps Using Topologically Consistent Triangulated Primitives **(Genus 1, 2, 3, 4 and 5)**. Please see **Appendix A.2** for the complete set of multi-view high-genus surface qualitative results.

5. Experiments and Results

Implementation details Our method leverages an adaptive remeshing algorithm developed in C++ using a half-edge-based data structure [3, 4, 16] to enable efficient local mesh operations. To balance computational efficiency and surface quality on a single 24GB RTX 3090, we set a frequency range of 130–200 iterations to periodically coarsen or refine the deforming mesh vertices $\mathbf{x} \in \mathbb{R}^{n\times 3}$; these updated mesh vertices are then passed to the current optimization loop, where they are rendered and optimized according to the l_1 rendering loss $\Phi(R(\mathbf{x}))$, implemented as a PyTorch extension using CUDA. To ensure timely reconstruction, we limit the optimization to 1500 iterations for low-genus surfaces and 3000 iterations for high-genus surfaces in both the baseline method and our approach. For multi-view reconstruction, we generate RGBA images of the ground truth surface by uniformly capturing [59] multi-view images with a radial distance away from a unit sphere to serve as our ground truth images, using a batch of 36 at a resolution of 1024×1024 for both low- and high-genus surfaces, and a batch of 120–360 at 256×256 in rare cases where the rendered views fail to capture essential topological features. Finally, we render the deforming mesh with vertices $\mathbf{x}$ using the same camera poses as those of the ground truth, allowing us to compute the rendering loss effectively. With this setup, the average reconstruction time across low-genus models is approximately 1 minute, while for high-genus models, it is approximately 2–5 minutes on a **single** 24GB NVIDIA RTX 3090 GPU.

5.1. Baselines and Evaluation Protocols

Baselines and Evaluation Dataset. Few existing works employ neural network-free approaches for surface *mesh-based* reconstruction in inverse rendering, primarily due to the difficulty of preserving surface topology as genus increases, introducing more tunnels and complex topological features. Many recent inverse rendering methods rely on alternative 3D representations. For example, Mehta et al.[44, 45] introduced level-set-based representations for smooth surface reconstructions, while Munkberg et al. [47] leveraged signed distance function (SDFs) within a neural framework to jointly optimize geometry, lighting, and materials. However, these methods are computationally expensive and not suitable for real-time applications. In contrast, surface *mesh-based* representations offer a balanced trade-off between efficiency and geometric detail but continue to struggle with topology preservation. To evaluate our method's ability to preserve surface topology in high-genus meshes, we compare against the state-of-the-art neural network-free approach by Nicolet et al. [51], recently integrated into Mitsuba-3 [24] and widely adopted as a strong baseline. Our evaluation includes five high-genus models, five genus-0 models from the Google Scanned Objects (GSO) dataset [10] and Stanford 3D Scanning Repository [63], as well as high-genus shapes from Gu et al. [16], Crane et al. [9], and the Thingi10K dataset [76].

Evaluation Metrics. We evaluate reconstruction quality using two standard metrics: Chamfer Distance (CD) and Intersection over Union (IoU). Following established practice, we first apply Iterative Closest Point (ICP) alignment between the reconstructed mesh and the ground-truth surface prior to computing these metrics, which assess the geometric accuracy of the reconstructed surface mesh.

5.2. Results

Multi-view Rendered Images. Figures 5 and 6 demonstrate that our method excels at reconstructing high-genus

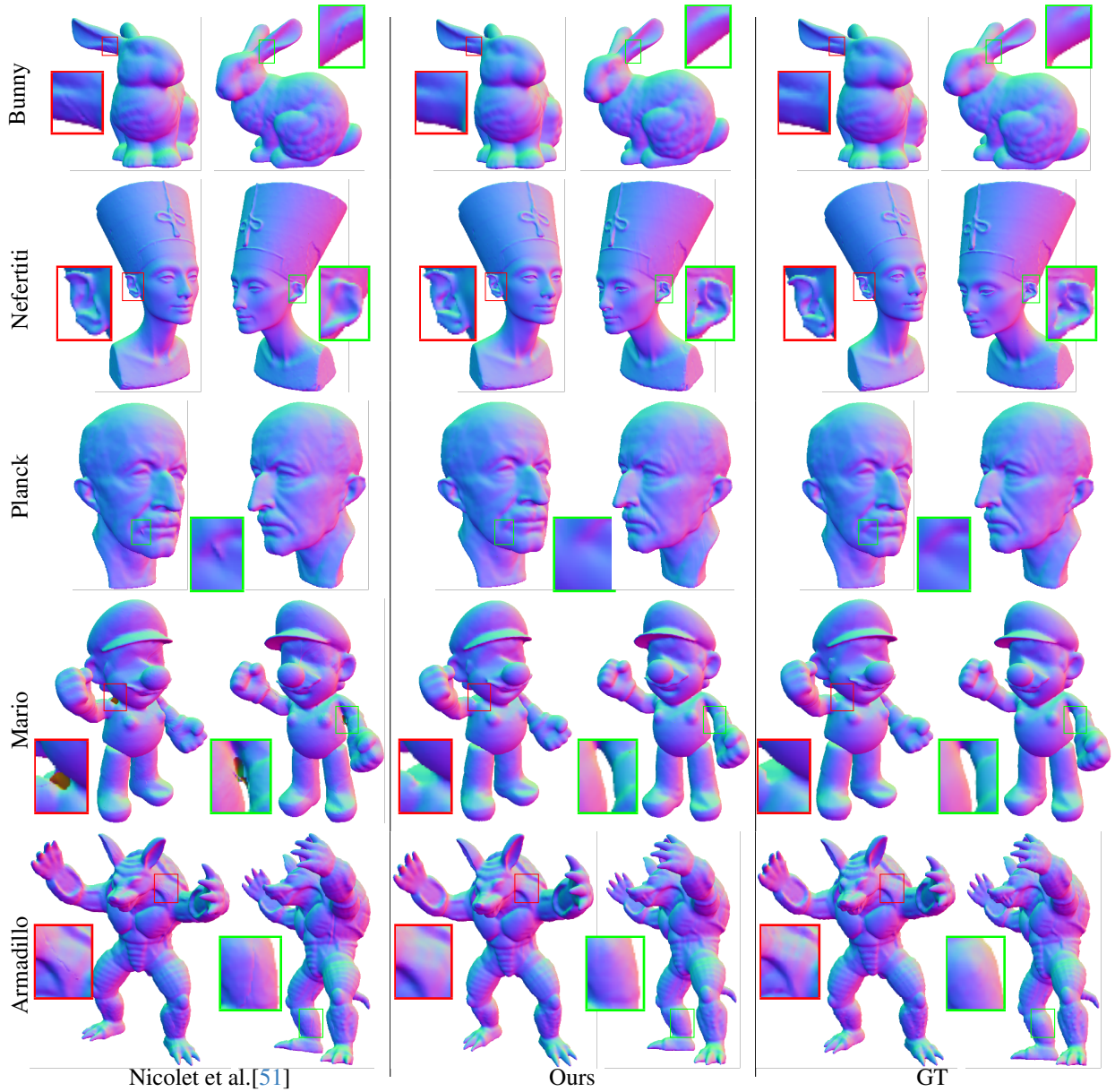

Figure 6. Qualitative Results of Multi-View Reconstruction for Genus 0 Surfaces with Rendered Views and Normal Maps Using a Sphere (**Genus 0**) as Topological Primitive. Please see **Appendix A.1** for the complete set of multi-view genus-0 surface qualitative results.

surfaces from multiple views, fully preserving topological features. In contrast, current state-of-the-art methods often fail to accurately capture these features, especially for high-genus surfaces. For low-genus surfaces, we show that our topologically informed approach enhances surface details and avoids visible seams, whereas existing methods frequently exhibit over-smoothing and cracks in regions of high curvature. These qualitative results highlight that relying solely on a reparameterized optimizer [51], without periodic coarsening and adaptive remeshing cycles, is insufficient for capturing key topological features. Tables 1 present quantitative comparison results, further demonstrating that our method excels at reconstructing high-genus surfaces while preserving topological features and enhancing surface detail for low-genus surfaces, as measured by Chamfer Distance and Volume IoU. Additionally, the genus numbers in the tables show that the genus of the reconstructed surfaces matches that of the ground truth, ensuring topological consistency. Figure 7 tracks the l_1 rendering loss $\Phi(R(\mathbf{x}))$ over optimization iterations, showing that our method generally converges in fewer iterations on average. For the Mario model, although the rendering loss plots converge to nearly zero, our method still achieves significantly better qualitative results. This demonstrates that

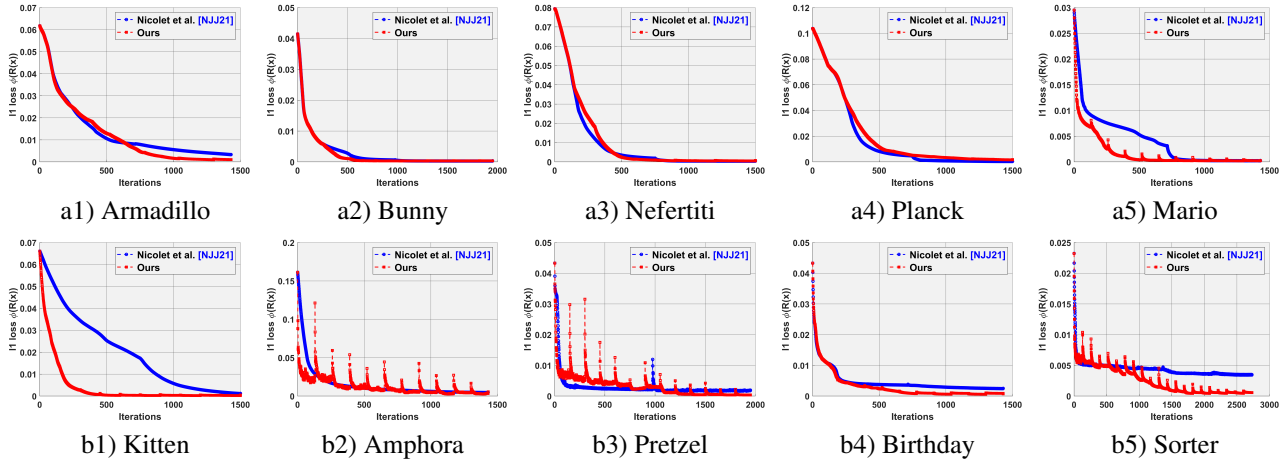

Figure 7. Rendering loss for low-genus surfaces (**Top Row**: Armadillo, Bunny, Nefertiti, Planck, Mario) and high-genus surfaces (**Bottom Row**: Kitten, Amphora, Pretzel, Birthday, Sorter) are shown, where **red** indicates our results and blue represents Nicolet et al. [51].

Table 1. Quantitative comparison with Nicolet et al. [51] for multi-view reconstructed low-genus and high-genus surfaces: Lower Chamfer Distance values means better result while higher Volume IoU means better result. **Please see Appendix D for full table.**

	Model	CD ↓		IoU ↑		$\chi(S)$	g
		[51]	**Ours**	[51]	**Ours**	**Ours**	**Ours**
Genus=0	Armadillo	0.0018	**0.0015**	0.8968	**0.9283**	2	0
	Bunny	0.0021	**0.0020**	0.8263	**0.8296**	2	0
	Nefertiti	0.0018	**0.0017**	0.8768	**0.9158**	2	0
	Planck	0.0019	**0.0018**	**0.9369**	0.9261	2	0
	Mario	**0.0024**	**0.0024**	0.8683	**0.9003**	2	0
Higher-Genus	Kitten	0.0039	**0.0025**	0.6298	**0.7126**	0	1
	Amphora	0.0054	**0.0033**	0.4581	**0.7924**	2	2
	Pretzel	0.0040	**0.0025**	0.6518	**0.8639**	4	3
	Birthday	0.0020	**0.0006**	0.4914	**0.8849**	6	4
	Sorter	0.0672	**0.0040**	0.2901	**0.7504**	8	5

a lower rendering loss does not necessarily correspond to a lower Chamfer Distance or a higher Volume IoU, while the reverse is generally true. The exception is the Planck model, where we set a limit of 1,500 iterations for evaluating low-genus surfaces. It requires more iteration to converge. Furthermore, Figure 8 demonstrates that our method better preserves surface features, including curvature and geometric continuity, whereas the current state-of-the-art often produces noisy curvature and discontinuities.

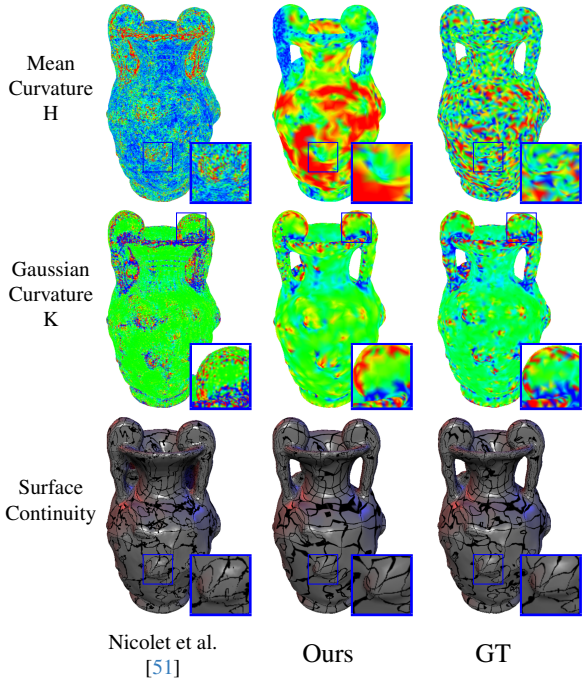

Figure 8. Qualitative comparison with Nicolet et al. [51] on surface quality on **Amphora (Genus-2)**. Less noisy curvature and more continuous reflection lines indicate higher surface quality.

6. Conclusion

We demonstrate that our topology-informed inverse rendering method can accurately reconstruct mesh surfaces from multi-view images, capturing essential topological features for high-genus surfaces while enhancing surface details for low-genus surfaces. Both qualitative and quantitative results show that our approach outperforms existing methods in achieving high topological accuracy and detail preservation, with superior performance in Chamfer Distance and Volume IoU, especially for high genus surfaces. We believe our method can provide valuable insights for generative tasks where topological consistency is crucial. Future work will focus on addressing complex topologies with intricate and narrow features, where illumination poses significant challenges, to further advance topology-aware inverse rendering in large-scale generative tasks.

References

[1] Dale Anderson, John C. Tannehill, Richard H. Pletcher, Ramesh Munipalli, and Viswanath Shankar. *Computational Fluid Mechanics and Heat Transfer*. CRC Press, 4th edition, 2020. 2

[2] Mario Botsch and Leif Kobbelt. An intuitive framework for real-time freeform modeling. *ACM Transactions on Graphics*, 23(3):630–634, 2004. 5

[3] Mario Botsch and Leif Kobbelt. A remeshing approach to multiresolution modeling. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 185–192. Association for Computing Machinery, 2004. 4, 5, 6

[4] Mario Botsch, Pierre Alliez, Bruno Lévy, Leif Kobbelt, and Mark Pauly. *Polygon Mesh Processing*. CRC Press, 2010. 2, 4, 5, 6

[5] Robert Bridson. *Fluid Simulation for Computer Graphics*. A K Peters/CRC Press, 2nd edition, 2015. 2

[6] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *arXiv*, 2021. 3

[7] Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[8] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[9] Keenan Crane, Fernando de Goes, Mathieu Desbrun, and Peter Schröder. Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 courses*, New York, NY, USA, 2013. ACM. 2, 4, 6

[10] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. *arXiv preprint arXiv:2204.11918*, 2022. 6

[11] Thomas Economon, Francisco Palacios, Sean Copeland, Trent Lukaczyk, and Juan Alonso. SU2: An Open-Source Suite for Multiphysics Simulation and Design. *AIAA Journal*, 54:1–19, 2015. 2

[12] Ziyang Fu, Yash Belhe, Haolin Lu, Liwen Wu, Bing Xu, and Tzu-Mao Li. Bsdf importance sampling using a diffusion model. In *SIGGRAPH Asia 2024 Conference Papers*. ACM, 2024. 2

[13] Xiang Gao, Xinmu Wang, Zhou Zhao, Junqi Huang, and Xianfeng David Gu. Hierarchical graphcut phase unwrapping based on invariance of diffeomorphisms framework. *IEEE Open Journal of Signal Processing*, 6:546–554, 2025. 2

[14] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. 2

[15] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3

[16] Xianfeng David Gu and Shing-Tung Yau. *Computational Conforaml Geometry*. High Education Press and International Press, 2008. 2, 4, 5, 6

[17] Minghao Guo, Bohan Wang, Kaiming He, and Wojciech Matusik. Tetsphere splatting: Representing high-quality geometry with lagrangian volumetric meshes. *arXiv preprint arXiv:2405.20283*, 2024. 2, 3

[18] Minghao Guo, Bohan Wang, Pingchuan Ma, Tianyuan Zhang, Crystal Elaine Owens, Chuang Gan, Joshua B. Tenenbaum, Kaiming He, and Wojciech Matusik. Physically compatible 3d object modeling from a single image. *arXiv preprint arXiv:2405.20510*, 2024. 2

[19] Kunal Gupta and Manmohan Chandraker. Neural mesh flow: 3d manifold mesh generationvia diffeomorphic flows, 2020. 2, 3

[20] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: A network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4):90:1–90:12, 2019. 2

[21] Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédo Durand. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG)*, 38(6): 201, 2019. 2

[22] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 2

[23] Alec Jacobson, Elif Tosun, Olga Sorkine, and Denis Zorin. Mixed finite elements for variational surface modeling. *Computer Graphics Forum*, 29(5):1565–1574, 2010. 5

[24] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. Mitsuba 3 renderer, 2022. https://mitsuba-renderer.org. 2, 3, 6

[25] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[26] Yucheol Jung, Hyomin Kim, Gyeongha Hwang, Seung-Hwan Baek, and Seungyong Lee. Mesh density adaptation for template-based shape reconstruction. In *ACM SIGGRAPH 2023 Conference Proceedings*. ACM, 2023. 2, 3

[27] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey, 2020. 2

[28] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 2, 3

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 3

[30] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (TOG)*, 39(6), 2020. 2, 3

[31] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics*, 37(6), 2018. 2

[32] Haixiang Liu, Yuanming Hu, Bo Zhu, Wojciech Matusik, and Eftychios Sifakis. Narrow-band topology optimization on a sparsely populated grid. *ACM Transactions on Graphics (TOG)*, 37(6), 2018. 2

[33] 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, 2023. 2

[34] 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. *arXiv preprint arXiv:2309.03453*, 2023.

[35] Zhen Liu, Yao Feng, Michael J. Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. In *International Conference on Learning Representations*, 2023.

[36] 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. *arXiv preprint arXiv:2310.15008*, 2023. 2

[37] Matthew M. Loper and Michael J. Black. Opendr: An approximate differentiable renderer. In *Computer Vision – ECCV 2014*, pages 154–169. Springer International Publishing, 2014. 2

[38] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987. 2

[39] Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Trans. Graph.*, 38(6), 2019. 2

[40] Fujun Luan, Shuang Zhao, Kavita Bala, and Zhao Dong. Unified shape and SVBRDF recovery using differentiable monte carlo rendering. *CoRR*, abs/2103.15208, 2021. 3

[41] Priyanka Mandikal, K L Navaneet, Mayank Agarwal, and R Venkatesh Babu. 3D-LMNet: Latent embedding matching for accurate and diverse 3d point cloud reconstruction from a single image. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018. 2, 3

[42] Steffen Marburg and Bodo Nolte. *Computational Acoustics of Noise Propagation in Fluids: Finite and Boundary Element Methods*. Springer, 2008. 2

[43] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Ieee/rsj International Conference on Intelligent Robots and Systems*, pages 922–928, 2015. 2

[44] Ishit Mehta, Manmohan Chandraker, and Ravi Ramamoorthi. A level set theory for neural implicit evolution under explicit flows. In *Proceedings of the 17th European Conference on Computer Vision (ECCV)*, pages 711–729. Springer, 2022. 2, 3, 6

[45] Ishit Mehta, Manmohan Chandraker, and Ravi Ramamoorthi. A theory of topological derivatives for inverse rendering of geometry, 2023. 2, 3, 6

[46] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 3

[47] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8280–8290, 2022. 3, 6

[48] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4), 2022. 2

[49] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia (GRAPHITE '06)*, pages 381–389, New York, NY, USA, 2006. Association for Computing Machinery. 4

[50] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2

[51] Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. Large steps in inverse rendering of geometry. *ACM Transactions on Graphics (TOG)*, 40(6):13, 2021. 1, 2, 3, 5, 6, 7, 8

[52] Werner Palfinger. Continuous remeshing for inverse rendering. *Computer Animation and Virtual Worlds*, 33(5):18, 2022. 2, 3

[53] D. Pardo, L. Demkowicz, C. Torres-Verdín, and M. Paszynski. A self-adaptive goal-oriented hp-finite element method with electromagnetic applications. part ii: Electrodynamics. *Computer Methods in Applied Mechanics and Engineering*, 196(37):3585–3597, 2007. Special Issue Honoring the 80th Birthday of Professor Ivo Babuška. 2

[54] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[55] Joseph E. Pasciak. The mathematical theory of finite element methods (susanne c. brenner and l. ridgway scott). *SIAM Review*, 37(3):472–473, 1995. 2

[56] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 2

[57] Ravi Ramamurti and William Sandberg. Simulation of flow about flapping airfoils using finite element incompressible flow solver. *AIAA Journal*, 39:253–260, 2001. 2

[58] Zhengyong Ren and Jingtian Tang. 3d direct current resistivity modeling with unstructured mesh by adaptive finite-element method. *Geophysics*, 75(1):H7–H17, 2010. 2

[59] Martin Roberts. How to evenly distribute points on a sphere more effectively than the canonical fibonacci lattice, 2023. 6

[60] Christoph Schwarzbach, Ralph-Uwe Börner, and Klaus Spitzer. Three-dimensional adaptive higher order finite element simulation for geo-electromagnetics – a marine csem example. *Geophysical Journal International*, pages no–no, 2011. 2

[61] Daniel Sieger and Mario Botsch. The Polygon Mesh Processing Library, 2023. 5

[62] Eftychios Sifakis and Jernej Barbic. Fem simulation of 3d deformable solids: A practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, pages 20:1–20:50, 2012. 2

[63] Stanford University Computer Graphics Laboratory. The stanford 3d scanning repository. Retrieved from http://graphics.stanford.edu/data/3Dscanrep/. 6

[64] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *arXiv*, 2023. 2

[65] Delio Vicini, Sébastien Speierer, and Wenzel Jakob. Differentiable signed distance function rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)*, 41(4):125:1–125:18, 2022. 2

[66] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems*, pages 27171–27183. Curran Associates, Inc., 2021. 2

[67] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis. *ACM Transactions on Graphics (SIGGRAPH)*, 36(4), 2017. 2

[68] Xinmu Wang, Xiang Gao, Xiyun Song, Heather Yu, Zongfang Lin, Liang Peng, and Xianfeng Gu. Ot-talk: Animating 3d talking head with optimal transportation. In *Proceedings of the 2025 International Conference on Multimedia Retrieval (ICMR '25)*, pages 1340–1349, New York, NY, USA, 2025. Association for Computing Machinery. 2

[69] Zichen Wang, Xi Deng, Ziyi Zhang, Wenzel Jakob, and Steve Marschner. A simple approach to differentiable rendering of sdfs. In *SIGGRAPH Asia 2024 Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. 2

[70] Zimo Wang, Cheng Wang, Taiki Yoshino, Sirui Tao, Ziyang Fu, and Tzu-Mao Li. Hotspot: Screened poisson equation for signed distance function optimization, 2024. 2

[71] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 2

[72] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *ICCV*, 2019. 3

[73] 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*, 128(12): 2919–2935, 2020. 3

[74] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. *arXiv*, 2019. 2

[75] Shuang Zhao, Wenzel Jakob, and Tzu-Mao Li. Physics-based differentiable rendering: from theory to implementation. *ACM Transactions on Graphics (TOG)*, page 30, 2020. 2, 3

[76] Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016. 6