

**M A S A R Y K
U N I V E R S I T Y**

FACULTY OF INFORMATICS

**Mesh Extraction from Radiance
Fields in Context of Transparent
Objects**

Master's Thesis

PETER KRAVÁR

Advisor: doc. RNDr. Barbora Kozlíková, Ph.D.

Department of Visual Computing

Brno, Spring 2025



Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Peter Kravár

Advisor: doc. RNDr. Barbora Kozlíková, Ph.D.

Acknowledgements

I would like to express my gratitude to doc. RNDr. Martin Madaras, PhD. and Mgr. Lukáš Gajdošech of the Skeletex team for their expert insights and valuable consultations throughout the course of this thesis. I am also deeply thankful to my thesis supervisor, doc. RNDr. Barbora Kozlíková, Ph.D., for her guidance and support particularly in shaping the formal aspects of this work.

Abstract

Methods based on radiance fields have continually pushed the boundaries of novel view synthesis since their emergence in 2020. Their capabilities have since been extended beyond view synthesis to high-quality surface reconstruction. However, these reconstruction methods generally perform poorly in scenes with transparent objects, often producing holes and artifacts in the resulting geometry. Reconstruction fails even in radiance fields that accurately represent transparent objects in novel view synthesis. This thesis investigates the causes of such failures and proposes a simple approach to mitigate them. The practical part builds on one such general purpose reconstruction method by compositing intermediate training outputs using segmentation, resulting in improved surface reconstruction for transparent objects.

Keywords

Gaussian Splatting, NeRF, radiance field, novel view synthesis, surface reconstruction, mesh extraction, transparent object

Contents

1	Introduction	1
2	Background	3
2.1	Terminology	3
2.1.1	Tasks	3
2.1.2	Methods	4
2.1.3	Terms	6
2.2	Novel View Synthesis	8
2.2.1	Neural Radiance Fields (NeRF)	9
2.2.2	3D Gaussian Splatting (3DGS)	11
2.3	Related Work	15
2.3.1	Novel View Synthesis	15
2.3.2	Radiance Fields and Surface Reconstruction	16
2.3.3	3D Segmentation of Radiance Fields	18
3	Theory	19
3.1	Mesh Extraction from Radiance Fields	19
3.1.1	NeRF	20
3.1.2	NeuS	20
3.1.3	SuGaR	23
3.1.4	2D Gaussian Splatting (2DGS)	26
3.1.5	Quadratic Gaussian Splatting (QGS)	27
3.1.6	PGSR	29
3.1.7	Gaussian Opacity Fields (GOF)	31
3.2	Reconstruction of Transparent Surfaces	35
3.2.1	Dex-NeRF	35
3.2.2	Residual-NeRF	36
3.2.3	GraspNeRF	36
3.2.4	Ref-NeRF	37
3.2.5	α Surf	38
3.2.6	Car-GS	41
3.2.7	TSGS	43
3.3	Chapter Summary	45
4	Method	47
4.1	Motivation and Overview	47

4.2	Baseline method analysis	48
4.3	Complete Method	50
4.4	Evaluation and Discussion	52
4.5	Additional Experiment: SH-Based Segmentation	56
5	Conclusion and Future Work	59
A	An appendix	61
	Bibliography	63

List of Figures

1.1	Transparent surface reconstruction: GOF and Ours	2
2.1	Novel View Synthesis showcase	8
2.2	NeRF pipeline overview	9
2.3	View-dependent effects captured by spherical harmonics	13
2.4	3DGS Adaptive density control	14
2.5	Open-Vocabulary segmentation by SuperGSeg	18
3.1	Logistic density $\phi_s(x)$ at different scales s	21
3.2	NeuS and NeRF surface reconstruction comparison	22
3.3	SuGaR Ablation study	25
3.4	Surface fitting comparison: ellipsoid, disk, quadric	27
3.5	Qualitative comparison: 2DGS and QGS	29
3.6	Unbiased depth estimation of PGSR	30
3.7	GOF: Definition of Gaussian’s normal	32
3.8	Qualitative comparison: PGSR, 2DGS, GOF	34
3.9	Dex-NeRF: grasping transparent objects.	36
3.10	Ref-NeRF appearance reparametrization	38
3.11	α Surf surface reconstruction qualitative comparison	41
3.12	Car-GS qualitative comparison	42
3.13	Transparency-depth dilemma	44
3.14	TSGS qualitative comparison	45
4.1	Gaussian origins, 30k and 1k2k3k4k	49
4.2	Segmentation process using SAGA.	51
4.3	Qualitative comparison: GOF and Ours, mesh	53
4.4	Qualitative comparison: GOF and Ours, normals	54
4.5	Qualitative comparison, transparent objects: GOF and Ours	55
4.6	Specular highlights captured by SH in GOF	57
4.7	Segmentation using SH coefficients	57
A.1	Ablations	62

1 Introduction

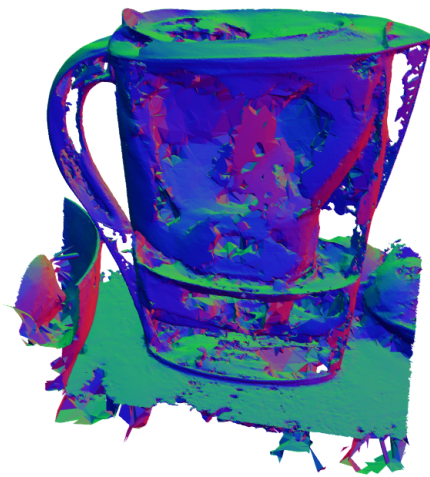
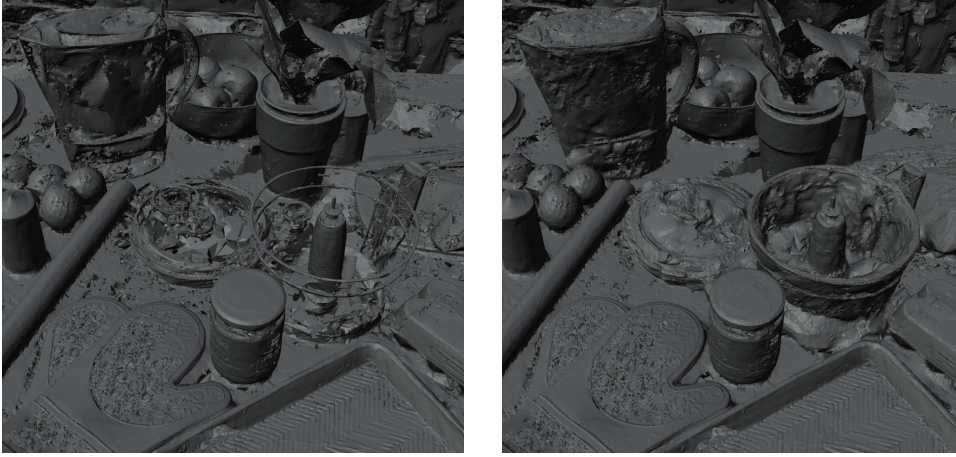
Radiance fields have emerged as a powerful framework for representing and rendering complex 3D scenes. These models describe the radiance emitted from any point in space along any direction, enabling the synthesis of novel views from previously unseen camera positions; a task known as novel view synthesis. Early approaches primarily employed implicit representations, most notably Neural Radiance Fields (NeRF) [37], while more recent work has adopted explicit representations, such as the one introduced by Gaussian Splatting (GS) [22]. The ability of radiance fields to capture both appearance and geometry makes them a compelling basis for surface reconstruction and mesh extraction, particularly in challenging scenarios involving transparent objects, where traditional techniques often fail to produce accurate geometry.

This thesis begins by introducing essential terminology and foundational concepts, with a focus on NeRF and GS, which form the basis of all methods discussed throughout the work. It then presents a critical analysis of key approaches to surface reconstruction from radiance fields, identifying recurring ideas and structural assumptions common across recent research. The final section of the theoretical part reviews specialized methods tailored for reconstructing scenes containing transparent objects.

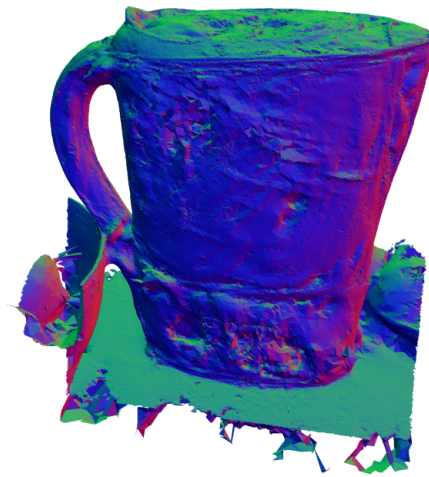
The practical contribution of the thesis proposes an effective extension to the Gaussian Opacity Fields method [63]. By compositing intermediate scene representations using segmentation, the proposed approach improves the quality of mesh extraction for transparent surfaces (see Fig 1.1).

Together, these components provide an up-to-date and analytically grounded perspective on surface reconstruction from radiance fields, with particular attention to the unique challenges posed by transparent geometry.

A related version of this work, focusing on the core ideas of the practical part, is to appear in the Proceedings of CESC 2025, Smolenice, Slovakia [28].



(a) GOF



(b) Ours

Figure 1.1: "Counter" scene of the Mip-NeRF 360 dataset [2] reconstructed using Gaussian Opacity Fields [63] and our improved version of their method. In GOF, transparent objects contain holes and artifacts. Our method improves reconstruction quality.

2 Background

A grasp of radiance fields in context of view synthesis is the prerequisite to understanding surface reconstruction based on these techniques. Therefore, this chapter aims to introduce the core terms and concepts from both areas, and provide a comprehensive overview of the related literature.

2.1 Terminology

The novelty of the topics addressed in this thesis necessitates the use of several technical terms, which are defined below. In many cases, these terms are used inconsistently or loosely across academic literature. Therefore, we support our definitions by citing representative works where the terms are employed in a consistent manner, even if not formally defined.

2.1.1 Tasks

Novel view synthesis is the task of generating 2D images of a 3D scene from previously unseen viewpoints. Given a set of input images of the scene, typically accompanied by camera intrinsics and extrinsics, a machine learning model is trained to learn the underlying scene structure and appearance. At inference time, the model takes a new camera pose as input and predicts the corresponding RGB image of the scene from that novel viewpoint [37].

Surface Reconstruction is the process of extracting a geometric surface, either in an explicit form (e.g., a mesh) or in an implicit form (e.g., a level set), from a 3D representation such as a neural radiance field [54] or a point-based opacity field [63].

Mesh Extraction is the specific process of converting a scene representation, such as a point cloud, signed distance field, or neural field, into a polygonal mesh [15].

2. BACKGROUND

2.1.2 Methods

This subsection outlines methods of computer graphics which are commonly employed in surface reconstruction.

Structure from Motion (SfM) recovers camera poses and a sparse 3D point cloud from a set of RGB images [45]. It detects and matches local features across overlapping image pairs, estimates relative camera geometry, and jointly optimizes all camera parameters and 3D points via bundle adjustment.

Input: RGB images

Output: Camera poses + sparse point cloud

Method: Feature detection and matching across views

Multi-View Stereo (MVS) generates a dense point cloud from RGB images and known camera poses (usually estimated by SfM). It computes per-image depth maps by matching corresponding pixels across multiple views, then fuses these maps into a coherent 3D representation [46].

Input: RGB images + camera poses

Output: Dense point cloud

Method: View-to-view feature matching and depth estimation

COLMAP is an open-source photogrammetry library that implements both Structure from Motion (SfM) and Multi-View Stereo (MVS). It provides a complete pipeline from unordered RGB images to a dense 3D point cloud and is widely used in academic and industrial 3D reconstruction tasks.

Input: RGB images

Output: Dense point cloud

Method: SfM & MVS

Gaussian Splatting is a real-time rendering technique that synthesizes 2D RGB images by projecting a set of 3D Gaussians from world space onto the image plane of a given camera pose. The projected Gaussians are depth-sorted, and their color contributions are accumulated using alpha blending to compute the final pixel colors [22].

Input: Gaussian point cloud, camera pose

Output: RGB image

Method: Projection to 2D plane, blending

In literature, "Gaussian splatting" may refer either to the rendering technique alone (as defined here), or to the full pipeline including optimization of a Gaussian point cloud. We use the narrower definition unless stated otherwise.

Truncated Signed Distance Fusion (TSDF Reconstruction) integrates depth maps from multiple views into a volumetric grid where each voxel stores a truncated signed distance to the nearest surface [9]. The accumulated field is converted into a surface using isosurface extraction methods, such as Marching Cubes.

Input: Depth maps (with known camera poses)

Output: Mesh

Method: Fuse distances into a TSDF volume, then extract surface

Marching Cubes is an algorithm for extracting a polygonal mesh from a 3D scalar field, such as a signed distance field or a density grid. It classifies the inside/outside state of voxel corners, consults a lookup table to determine how the surface intersects the voxel, and generates triangles accordingly [34].

Input: 3D scalar field (e.g., SDF, TSDF, density grid)

Output: Mesh

Method: Lookup-based triangulation of voxel surface intersections

Marching Tetrahedra is a topology-safe surface extraction algorithm that operates on tetrahedral cells instead of cubes. It linearly interpolates scalar values at the vertices of each tetrahedron and applies a fixed lookup rule to generate triangles [50]. While originally used with tetrahedra derived from regular voxel grids, the method is also applicable to irregular tetrahedral meshes, such as those used in adaptive reconstruction pipelines.

Input: 3D scalar field (e.g., SDF, TSDF, density grid) defined over a tetrahedral mesh
Output: Mesh
Method: Tetrahedral cells are processed for unambiguous, lookup-based triangulation.

Poisson Surface Reconstruction produces a watertight mesh from a dense point cloud with oriented normals [21]. It formulates surface reconstruction as a spatial Poisson problem, which it then solves for a function whose gradient matches the input normals, and extracts the surface as the zero level set of this function.

Input: Oriented dense point cloud (depth + normal maps)
Output: Mesh
Method: Solves a Poisson equation based on normal integration

2.1.3 Terms

Transparent Object is any object present in a 3D scene which displays at least partial transparency.

Radiance Field is the representation of a 3D scene, which describes the color and intensity of light passing through any point in space along any direction [37]. It is typically modeled as a five-dimensional function that maps a 3D position and 2D viewing direction to an RGB color:

$$L(x, y, z, \theta, \phi) \rightarrow (r, g, b).$$

By evaluating the radiance along pixel rays of a virtual camera, the radiance field can be rendered into a 2D image.

Neural Radiance Field (NeRF) also commonly referred to as "vanilla NeRF", or "the original NeRF" [37], is an implicit representation of a radiance field, where the color and density of a 3D scene are modeled by a neural network, typically a multilayer perceptron. Given a 3D point and a viewing direction as input, the network predicts the corresponding RGB color and volumetric density, enabling photorealistic novel view synthesis through volume rendering.

Spherical Harmonics are a set of orthonormal basis functions defined on the surface of a sphere. They are commonly used to represent arbitrary view-dependent functions over spherical domains, such as lighting or reflectance [12]. Each SH basis function is denoted as Y_l^m , where l is the degree (controlling level of detail) and m is the order, such that $-l \leq m \leq l$ (controlling the orientation and shape).

3D Gaussian point cloud is an explicit representation of a radiance field, where the scene is modeled as a set of anisotropic 3D Gaussians [35]. Each Gaussian is defined by a spatial mean, a full 3D covariance matrix (encoding shape and orientation), spherical harmonics coefficients for view-dependent color, and a scalar density.

2D Gaussian point cloud is an explicit representation of a radiance field, where the scene is modeled as a set of anisotropic two-dimensional Gaussians residing in 3D world space [18]. The definition of Gaussian primitives remains the same as in the case of 3D Gaussian point cloud.

Signed Distance Function (SDF) is a 3D scalar field where each point stores the shortest distance to the surface of an object. The sign of the value indicates whether the point lies inside (negative) or outside (positive) the surface, and the zero level set defines the surface itself [13].

Truncated Signed Distance Function (TSDF) is a practical variant of the signed distance function used in volumetric reconstruction. Distance

2. BACKGROUND

values are truncated to a limited range near the surface, which improves robustness to noise and enables efficient fusion of depth data from multiple views into a smooth surface.

Photometric loss is a scalar number that measures the discrepancy between a rendered image and its corresponding ground truth image [37]. It is widely used as a training objective in differentiable rendering, and is typically computed using metrics such as the L1 norm, L2 norm, or the Structural Similarity Index Measure (SSIM).

2.2 Novel View Synthesis

The concepts presented in this section form the theoretical foundation for the remainder of this thesis. Radiance fields are mainly used in novel view synthesis (see Fig. 2.1), a task that, while challenging, is comparatively more tractable than surface reconstruction. As such, it provides a natural starting point for understanding the principles behind radiance-based 3D representations. In subsequent sections, we build upon this foundation to explore how these representations can be extended and adapted for surface reconstruction.



(a) Mip-NeRF 360 (Implicit)

(b) 3DGS (Explicit)

Figure 2.1: Example of an implicit and an explicit radiance field-based method for novel view synthesis. The methods learn a 3D representation from ground truth images, and predict images from unseen view-points. Both shown images are synthesized. Image source: 3DGS [22]

2.2.1 Neural Radiance Fields (NeRF)

NeRF in its original form [37] utilizes three main components to perform photorealistic novel view synthesis:

Multilayer perceptron (MLP) is trained to approximate the function

$$F : \mathbb{R}^3 \times \mathbb{S}^2 \rightarrow \mathbb{R}^3 \times \mathbb{R}, \quad (\mathbf{x}, \mathbf{v}) \mapsto (\mathbf{c}, \sigma), \quad (2.1)$$

where $\mathbb{S}^2$ denotes the unit sphere in $\mathbb{R}^3$: $\mathbb{S}^2 = \{\mathbf{v} \in \mathbb{R}^3 \mid \|\mathbf{v}\| = 1\}$.

The function is commonly expressed as $(x, y, z, \theta, \phi) \rightarrow (r, g, b, \sigma)$, where (x, y, z) specifies the 3D point $\mathbf{x}$ and (θ, ϕ) the viewing direction $\mathbf{v}$. The MLP outputs (r, g, b) as the emitted color $\mathbf{c}$ and volumetric density σ at that location. To ensure consistent density estimation across views, the network is divided into two components. The geometry network predicts density solely as a function of spatial position, independent of viewing direction. The appearance network then conditions on both position and viewing direction to predict view-dependent color. This learned volumetric representation forms the basis for computing the radiance field via differentiable volume rendering.

Differentiable volume rendering is used to obtain 2D images from an implicit 3D representation; in this case an MLP. Rays are cast into the 3D scene from a virtual camera. Color and opacity contributions of the scene are accumulated along the rays to render images, and to calculate the photometric loss for MLP training (see Fig. 2.2).

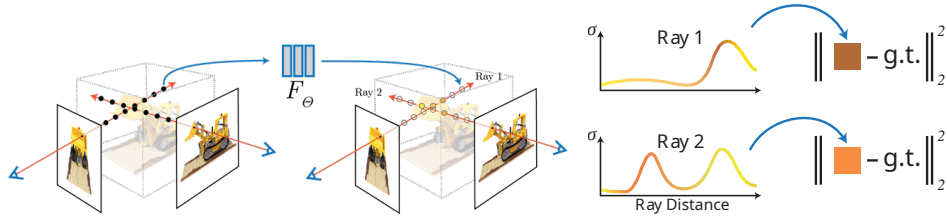


Figure 2.2: NeRF pipeline overview, image from Mildenhall et al. [37].

2. BACKGROUND

The expected color $C(\mathbf{r})$ of a camera ray is:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t) \sigma(\mathbf{r}(t)) \mathbf{c}(\mathbf{r}(t), \mathbf{d}) dt, \quad (2.2)$$

where $T(t) = \exp \left(- \int_{t_n}^t \sigma(\mathbf{r}(s)) ds \right),$

with the following variable meanings:

- $\mathbf{r}(t)$: Ray from origin $\mathbf{o}$ in direction $\mathbf{d}$: $\mathbf{o} + t\mathbf{d}$,
- t : Distance along the ray,
- t_n, t_f : Near and far bounds of ray integration,
- $\sigma(\mathbf{r}(t))$: Volume density at point $\mathbf{r}(t)$, representing opacity,
- $\mathbf{c}(\mathbf{r}(t), \mathbf{d})$: Emitted color at point $\mathbf{r}(t)$ in direction $\mathbf{d}$,
- $T(t)$: Accumulated transmittance from t_n to t .
Intuitively: probability of no occlusion.

The volume rendering integral in Equation 2.2 is approximated by discretizing the ray into N sampled points. The accumulated color $\hat{C}(\mathbf{r})$ is computed as:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i, \quad (2.3)$$

where

$$T_i = \exp \left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right),$$

$$\delta_i = t_{i+1} - t_i.$$

Here, δ_i is the spacing between consecutive samples, and T_i represents the accumulated transmittance up to point i .

NeRF trains its MLP by minimizing the discrepancy between the accumulated color $\hat{C}(\mathbf{r})$ of each ray and the ground truth pixel color from the corresponding input image.

Positional encoding transforms scalar inputs into a space of L dimensions $\gamma : \mathbb{R} \rightarrow \mathbb{R}^{2L}$. The transformation is described by the following function:

$$\gamma(p) = \left(\sin(2^0 \pi p), \cos(2^0 \pi p), \dots, \sin(2^{L-1} \pi p), \cos(2^{L-1} \pi p) \right).$$

Positional encoding mitigates the spectral bias of neural networks toward low-frequency functions [42] and enables learning of fine details, as highlighted by the ablation study included in the NeRF paper [37]. In NeRF, positional encoding is applied to every scalar of the input.

2.2.2 3D Gaussian Splatting (3DGS)

This subsection describes the full pipeline proposed in the paper “*3D Gaussian Splatting for Real-Time Radiance Field Rendering*” by Kerbl et al. [22]. Their method encompasses both the real-time rendering technique and the optimization process used to construct the Gaussian-based radiance field from input images. This short explanation places emphasis on those parts of the method, which are relevant for techniques presented later.

3D Gaussian primitive is based on an equation, which describes the spatial influence of an anisotropic Gaussian centered at the origin with covariance matrix Σ on a point in space $x \in \mathbb{R}^3$ [65]:

$$G(x) = e^{-\frac{1}{2}x^\top \Sigma^{-1}x}. \quad (2.4)$$

In addition to the covariance matrix, Gaussian primitives are defined by the following parameters:

- $\mu \in \mathbb{R}^3$: 3D mean position of the Gaussian,
- $\Sigma \in \mathbb{R}^{3 \times 3}$: Symmetric positive-definite covariance matrix controlling shape and orientation,
- $\alpha \in \mathbb{R}$: Opacity (alpha) value,
- $\mathbf{c}_{\text{sh}} \in \mathbb{R}^{3 \times B}$: Spherical Harmonics coefficients for view-dependent color, where B is the number of SH basis functions.

2. BACKGROUND

To ensure that the covariance matrix Σ has a valid geometric interpretation, it must be positive semi-definite. However, directly enforcing this constraint during training is non-trivial. Instead, the covariance is represented via a scaling matrix S and a rotation matrix R . These transformation matrices are composed to form the covariance as:

$$\Sigma = RSS^\top R^\top. \quad (2.5)$$

In practice, each Gaussian primitive stores two parameters to represent this arrangement: a scaling vector $\mathbf{s} \in \mathbb{R}^3$, and a quaternion $\mathbf{q}$ encoding the rotation.

The opacity parameter α is passed through a sigmoidal activation function before evaluation, effectively restricting it to the range $[0, 1]$. Similarly, the covariance scaling vector is transformed via an exponential activation prior to constructing the matrix. Both transformations promote smoother gradients during training.

Spherical Harmonics are used to model view-dependent color effects such as specular highlights and reflections in Gaussian primitives. Instead of storing a single RGB color per Gaussian, the color $\mathbf{c}(\mathbf{v})$ is expressed as a function of the viewing direction $\mathbf{v}$ using Spherical Harmonics (typically order 3). Each Gaussian stores a set of SH coefficients:

$$\mathbf{c}(\mathbf{v}) = \sum_{l=0}^L \sum_{m=-l}^l \mathbf{c}_{lm} Y_{lm}(\mathbf{v}),$$

where $\mathbf{c}_{lm} \in \mathbb{R}^3$ are the learned RGB coefficients and $Y_l^m(\mathbf{v})$ are the real spherical harmonics basis functions evaluated at direction $\mathbf{v}$. This formulation enables efficient encoding of complex lighting behavior, as shown by Figure 2.3.

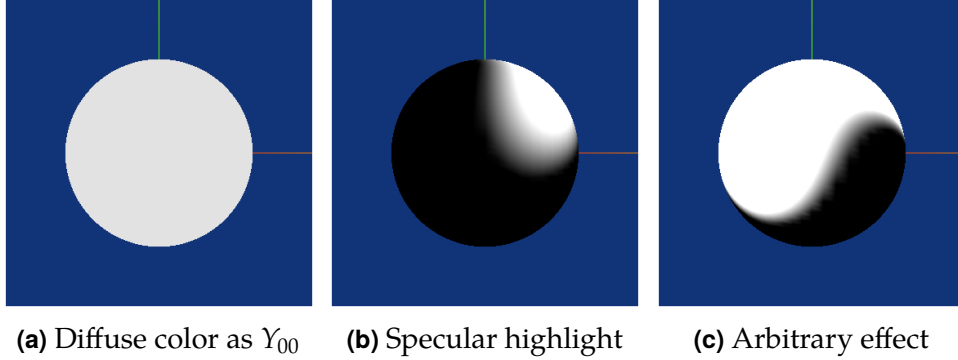


Figure 2.3: Sphere appearance encoded as linear combinations of SH basis functions up to second degree.

Splatting is a rendering technique, in which volume elements are projected into screen space as 2D kernels and composited using alpha blending to produce an image [55]. Zwicker et al. [65] show how to perform this for 3D Gaussian primitives. Given a viewing transformation W , the covariance matrix Σ' of a 3D Gaussian in camera space can be obtained as:

$$\Sigma' = JW\Sigma W^T J^T, \quad (2.6)$$

where J is the Jacobian of the affine approximation of the projective transformation.

The resulting 2D screen-space Gaussian encodes the distribution of opacity projected onto the image plane in an elliptical footprint. To render a full Gaussian point cloud, all primitives are splatted and sorted by depth. Each splat is rasterized with per-pixel weights determined by Equation 2.4. Finally, color contributions are composited using alpha blending. This process is differentiable, and can thus be directly optimized.

The training pipeline first initializes the Gaussian primitives as isotropic Gaussians centered at the points of an SfM point cloud. The scene is then iteratively rendered from camera viewpoints which match ground truth images, and the Gaussian features are optimized based on photometric loss via gradient descent. An adaptive density con-

2. BACKGROUND

control strategy is employed to prune and clone Gaussians in necessary regions (see Fig. 2.4).

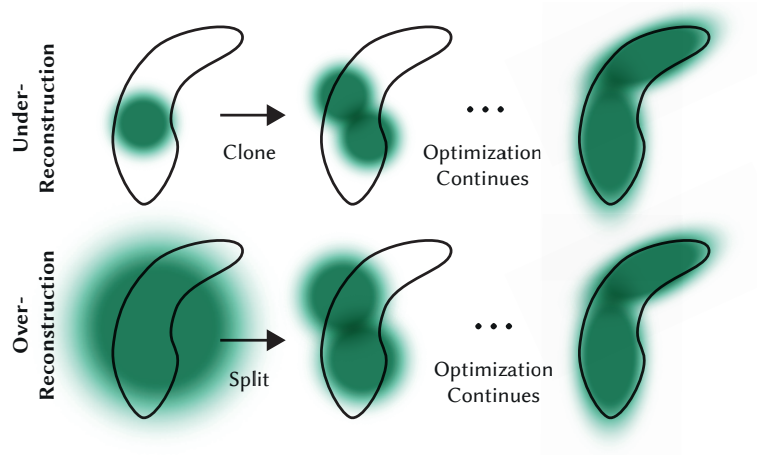


Figure 2.4: Adaptive density control of 3D Gaussian primitives. During cloning, the means of the new pair of Gaussians are initialized by sampling the original Gaussian as a PDF; scale factor is reduced by an experimentally determined amount. Image source: Kerbl et al. [22].

2.3 Related Work

This section begins with an overview of early radiance fields methods invented for view synthesis, and summarizes subsequent developments which established the research field. The following subsection explains how the capabilities of radiance fields were extended from pure view synthesis to surface reconstruction, and introduces improvements proposed specifically for scenes with transparent objects. The final subsection discusses the existing techniques for segmentation of radiance fields.

2.3.1 Novel View Synthesis

Neural Radiance Fields (NeRF) [37], in which the authors utilized a multilayered perceptron (MLP) [16] to learn a 3D scene by optimizing photometric loss through volume rendering, emerged as a critical innovation in the area of novel view synthesis.

The original NeRF implementation was computationally demanding for both training and rendering. Hence, a major line of research focused on improving viewing performance by baking the NeRF into different representations after training [62, 17]. Poor training performance was addressed by the later works of Plenoxels [12] and DVGO [48], which showed that the MLP in NeRF was being used as an inefficient volumetric data structure and could be replaced by a voxel grid. While voxel grids do achieve significantly faster inference than NeRF, they consume large amounts of memory and struggle to represent fine details due to their limited resolution. Nonetheless, their emergence sparked a broader search for more efficient and expressive data structures for radiance field representation. Later approaches attempted to replace the MLP with a wide variety of data structures; even large amounts of smaller MLPs [44] and 4D tensors [4]. Meanwhile, a major breakthrough in NeRF performance was achieved by multiscale hash encoding in InstantNGP [38], yet the search for an efficient data structure continued and trended towards increasingly more explicit representations. Such representations include neural point clouds [59], and finally Gaussian point clouds.

3D Gaussian Splatting [22] achieves real-time image synthesis by projecting 3D Gaussians into 2D screen space, sorting them by depth,

2. BACKGROUND

and compositing their contributions, all while pertaining the ability to optimize a scene directly. Their work sparked a surge of follow-up research. Among others, further works focused on addressing performance [36], removing rendering artifacts [41], and reducing sensitivity to initialization conditions and local minima [24]. Unlike NeRF, which relies on computationally intensive ray marching, Gaussian Splatting offers a more efficient and practical rendering pipeline. Its compatibility with standard 3D software and strong empirical performance have made Gaussian-based representations a focus of recent research. Nevertheless, implicit methods like NeRF remain highly relevant: they introduced many core ideas and mechanisms, such as differentiable volumetric rendering and learned view-dependent appearance, which continue to influence explicit approaches. As such, understanding both paradigms provides a more complete view of the field.

2.3.2 Radiance Fields and Surface Reconstruction

NeRF allows for depth maps to be obtained for free as a byproduct of rendering even in the original implementation [37]. The problem of surface reconstruction can therefore be reduced to mesh extraction from depth maps by applying a TSDF mesh extraction algorithm [39]. However, later methods highlight numerous issues with this naive approach and commonly use supplemental implicit abstractions to represent the surface prior to mesh extraction in order to mitigate surface artifacts [17, 40, 54, 61, 43]. Despite some NeRF-based surface reconstruction methods achieving impressively high detail [32], they commonly require upwards of 12 hours of training per scene on an expensive GPU, such as the NVIDIA A100.

3D Gaussian Radiance Fields make geometry extraction more challenging due to their explicit nature and unclear surface boundaries. SuGaR [15] mitigates the problem of unclear surface boundary by introducing regularization terms into the training pipeline. The regularization terms constrain Gaussians to form more consistent and flat surfaces, allowing for estimation of a normal as the shortest axis for any given Gaussian. The normal information enables the use of Poisson mesh extraction, yielding meshes of higher quality. The approach of using flat Gaussian primitives to enhance surface reconstruction

was largely successful and appeared in numerous other notable techniques, such as 2DGS [18] and PGSR [5]. An alternative paradigm for surface extraction from Gaussian point clouds was introduced by Gaussian Opacity Fields (GOF) [63]. Unlike methods that modify the representation, GOF operates directly on standard 3D Gaussian primitives. It constructs an opacity field from the accumulated contributions of the Gaussians and extracts surfaces as level sets of this field. Their approach eliminates the need for Poisson surface reconstruction and enables more efficient and scalable surface extraction, particularly in unbounded scenes.

Numerous other methods aim to guide the surface reconstruction process using priors and multimodal data [51, 56, 53, 29], such as ground truth normal maps obtained with an accurate sensor.

GS-based surface reconstruction methods are up to two orders of magnitude faster than implicit approaches, enabling training within tens of minutes even on a single high-end consumer-grade GPU [15, 5].

Surface reconstruction of transparent objects is a largely underexplored problem in the context of radiance fields. Several methods have attempted to leverage NeRFs [19, 23, 10, 11] and Gaussian Splatting [1, 25] for depth estimation and downstream tasks such as robotic grasping of transparent objects. However, these approaches are limited to producing imprecise depth maps and are not capable of generating explicit surface meshes.

True surface reconstruction was attempted by few methods, each of which introduces specific limitations. NeRRF [7] requires object silhouette as an additional input. Nu-NeRF [49] is computationally demanding. α Surf [57], despite its very impressive results, builds on the voxel-based representation of Plenoxels [12] and thus inherits its associated limitations in scene resolution and memory efficiency. To the best of our knowledge, the only Gaussian-based methods that have demonstrated success in reconstructing transparent surfaces under unconstrained conditions are Car-GS [30], which focuses specifically on car models, and TSGS [31]. As of the time of writing, the implementations of both methods are yet to be released.

2.3.3 3D Segmentation of Radiance Fields

Segmentation of radiance fields is desirable in numerous problems surrounding view synthesis.

Most existing NeRF-based segmentation methods work in image space by utilizing the SAM segmentation model [26] to obtain semantic maps from ground truth images. The semantic maps are then extrapolated into 3D by training a separate volumetric representation.

Most Gaussian-based methods use the SAM segmentation maps to train additional 3D Gaussian features [3] for existing Gaussian point clouds. Runtime segmentation can be then guided by user input in the form of one-shot click prompts [8, 3], or open-vocabulary prompts [33, 6, 58] as shown in Figure 2.5.

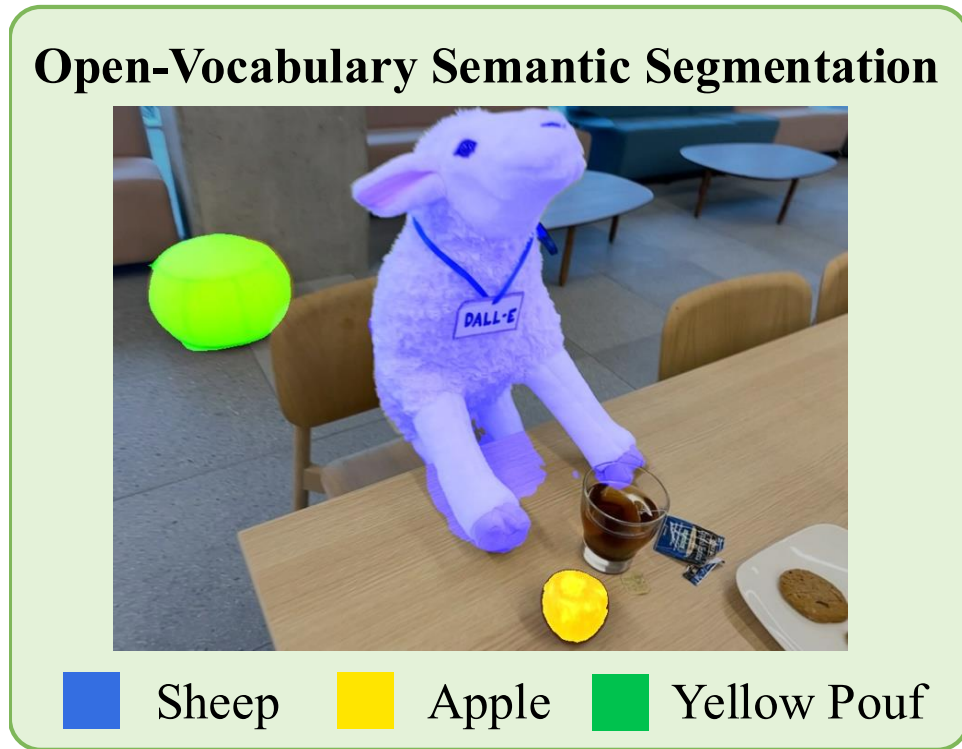


Figure 2.5: Open-Vocabulary segmentation by SuperGSeg [33]. Gaussian primitives are segmented based on text prompts. Figure adopted from SuperGSeg [33].

3 Theory

Radiance fields are a very active topic in computer graphics research. Thorough explanation of all the concepts introduced by novel approaches is beyond the scope of this thesis. Therefore, this chapter selectively explains and compares the most relevant theoretical concepts introduced by seminal papers, with emphasis on those still used in state of the art methods for surface reconstruction.

3.1 Mesh Extraction from Radiance Fields

This section introduces a selection of surface reconstruction methods based on radiance fields capable of producing mesh outputs. Rather than going into full technical detail, each method is described in a concise way, focusing on the core ideas and high-level concepts which they introduce.

Surface reconstruction using radiance fields is a nascent subfield within the rapidly evolving area of radiance field research. At this stage, there is no clearly established superior method. Instead, many different approaches are being explored in the research community, with state of the art methods frequently changing.

Newer methods often achieve better results on benchmark datasets, but this does not mean they are fundamentally better in terms of core ideas. In many cases, performance gains come from addressing practical issues already well known in traditional 3D graphics, such as exposure handling. Conversely, some methods may report lower benchmark scores simply because they overlook such implementation details, or because they predate newer advances such as more effective scene optimization algorithms. For these reasons, methods presented in this section were chosen based on their lasting impact on the field of surface reconstruction, rather than benchmark scores.

It is worth noting that a large body of work exists on surface reconstruction using NeRF-based approaches [54, 40, 32]. However, many of these methods are impractical for real-world use due to their extreme computational demands. Despite significant advances made in accelerating training speeds of standard NeRF models [38], these acceleration techniques are rarely adopted by surface reconstruction

3. THEORY

techniques. They complicate the implementation and introduce minor artifacts undesirable in methods which strive to push state of the art of surface reconstruction quality. As a result, many NeRF-based surface reconstruction pipelines rely on slow optimization procedures that can take many hours on datacenter-grade GPUs, making them impractical in real world applications. For the aforementioned reasons, this section puts more emphasis on the explicit radiance fields. Nonetheless, some NeRF-based approaches are included for their important conceptual contributions which contrast and complement the design philosophies of explicit methods.

3.1.1 NeRF

In neural radiance fields (NeRF), a depth map can be obtained alongside the RGB output by taking the expected termination distance of each ray, computed as the weighted average of sampled depths along the ray, weighted by the transmittance and density. These depth maps can be used as input for TSDF fusion algorithms to obtain mesh output.

Alternatively, the voxel-based marching cubes algorithm can be applied directly to the density field represented by the NeRF’s multi-layer perceptron (MLP). A surface is extracted by applying marching cubes to a voxel grid where each voxel stores the accumulated density. Voxels with density values above a chosen threshold are considered to be part of the surface.

3.1.2 NeuS

NeuS [54] addresses a limitation of NeRF: the photometric loss does not encourage sharp surface boundaries. The learned density field tends to produce thick, fuzzy surfaces, which causes noisy depth maps and complicates mesh extraction.

Instead of modeling opacity directly, NeuS represents the scene as an implicit surface via a signed distance function (SDF). It introduces two functions:

- $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ maps a 3D position $\mathbf{x} = (x, y, z)$ to its signed distance to the surface.

- $c : \mathbb{R}^3 \times \mathbb{S}^2 \rightarrow \mathbb{R}^3$ maps a spatial position and view direction $\mathbf{v} = (v_1, v_2, v_3) \in \mathbb{S}^2 \subset \mathbb{R}^3$ to RGB color.

Both are represented by separate MLPs. Notice their correspondence to the geometry and appearance network modules introduced in NeRF, discussed in Section 2.2.1.

The surface is defined implicitly as the zero level set:

$$S = \{\mathbf{x} \in \mathbb{R}^3 \mid f(\mathbf{x}) = 0\}.$$

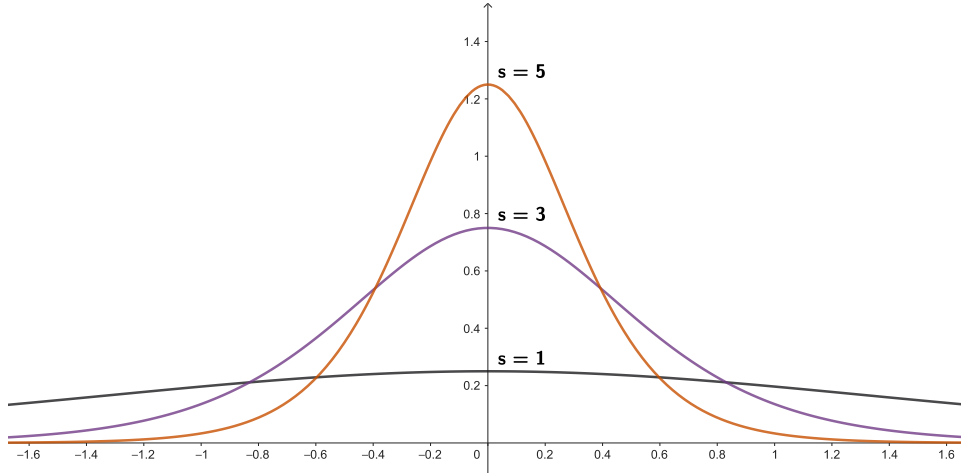


Figure 3.1: Logistic density $\phi_s(x)$ at different scales s .

To enable volume rendering, NeuS transforms the SDF into a density field using the smooth, unimodal logistic probability density function (see Fig. 3.1)

$$\phi_s(f(\mathbf{x})) = \frac{se^{-sf(\mathbf{x})}}{(1 + e^{-sf(\mathbf{x})})^2}.$$

While the exact expression is not critical, it is important to note that the sharpness is controlled by a parameter s , and the network learns $1/s$ as a trainable scalar. As training progresses, $1/s$ naturally decreases. This sharpens the surface density and allows the network to overfit the surface. This behavior is desired; the rendering loss concentrates on the zero level set, pulling the SDF tightly around the

3. THEORY

true geometry and allowing the model to represent high-frequency surface detail.

Training proceeds similarly to NeRF, using volume rendering and photometric loss on RGB ground truth images with known camera poses. For mesh extraction from the produced density field, NeuS uses the marching cubes algorithm.

For NeuS results and a qualitative comparison with NeRF, refer to Figure 3.2.

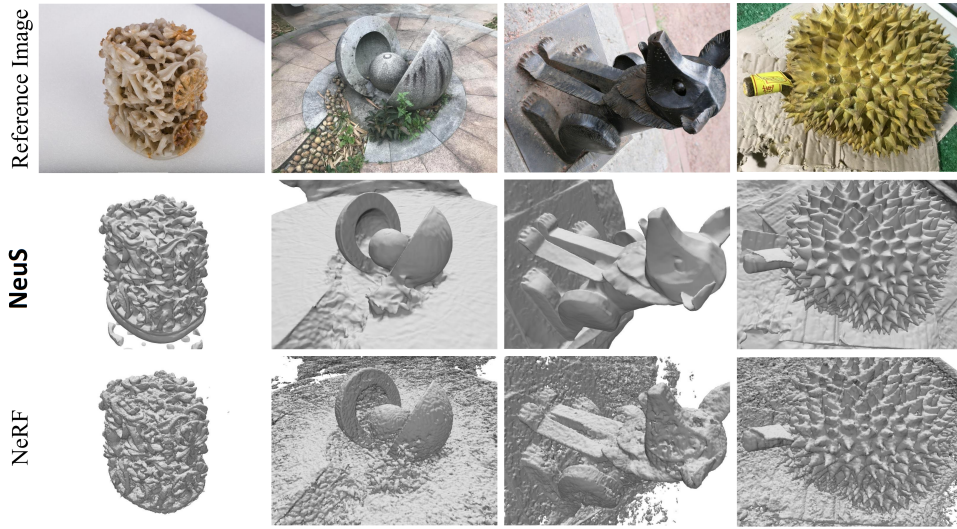


Figure 3.2: NeuS and NeRF surface reconstruction comparison on the DTU MVS dataset [20]. Images taken from NeuS [54].

3.1.3 SuGaR

SuGaR [15] is a GS-based surface reconstruction method which was long considered the state of the art. It inspired an entire family of surface reconstruction approaches based on planar Gaussians.

Surface Regularization is the key idea presented by SuGaR. In the training process, a newly introduced regularization term encourages the Gaussians to adhere closely to the underlying scene surfaces. This results in a Gaussian representation that more accurately captures the true geometry.

Specifically, the regularization enforces the following properties:

1. **Limited Overlap:** The density at any point should be estimable from a single Gaussian. This promotes the spatial spreading of Gaussians across the surface and discourages excessive overlap.
2. **Flatness:** Gaussians are encouraged to be flat by constraining one of their scaling axes to approach zero, effectively aligning each Gaussian with the surface it represents.
3. **Opacity Binarization:** Gaussians are driven toward being either fully opaque or fully transparent, enabling the removal of non-contributing Gaussians and improving the compactness of the representation.

Density Representation: In any generic 3D Gaussian point cloud, the Gaussians collectively define a density field that reflects the scene structure. The density at a point $p \in \mathbb{R}^3$ can be computed as:

$$d(p) = \sum_g \alpha_g \exp \left(-\frac{1}{2} (p - \mu_g)^T \Sigma_g^{-1} (p - \mu_g) \right), \quad (3.1)$$

where α_g is the opacity, μ_g is the center, and Σ_g is the covariance matrix of Gaussian g .

However, when the aforementioned regularization conditions hold, the density can be approximated more simply as:

$$\bar{d}(p) = \exp \left(-\frac{1}{2s_{g^*}^2} ((p - \mu_{g^*}) \cdot n_{g^*})^2 \right), \quad (3.2)$$

3. THEORY

where g^* is the Gaussian most influencing p , n_{g^*} is its normal vector, and s_{g^*} is its scaling along the normal direction.

In this approximation, each surface point p is primarily influenced by a single fully opaque, flat Gaussian. The summation disappears due to the lack of overlap, and the 3D quadratic form simplifies to a dot product, capturing only the displacement along the normal direction.

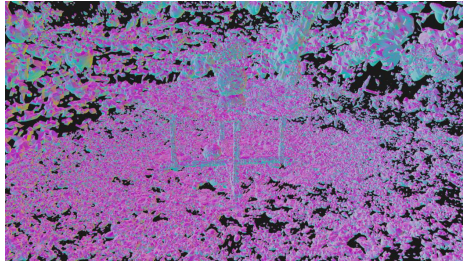
Surface regularization can thus be encouraged by minimizing the quantity $|d(p) - \bar{d}(p)|$ as a regularization loss. Although SuGaR employs a slightly more sophisticated SDF-based term in practice, the underlying principle in which they apply regularization to improve surface reconstruction remains the same.

Mesh extraction methods such as marching cubes are not well-suited to the Gaussian-induced density field (see Fig. 3.3b). The learned Gaussians vary significantly in size and shape to capture both coarse and fine structures, leading to an opacity field that is nearly zero in most of space, with extremely sharp transitions near the surface. Since marching cubes requires a smooth scalar field to generate high-quality meshes, it fails under these conditions. The approach of using a Poisson reconstruction on the point cloud of Gaussian origins with estimated normals also fails (see Fig. 3.3c) due to gaps between large Gaussian primitives being too wide.

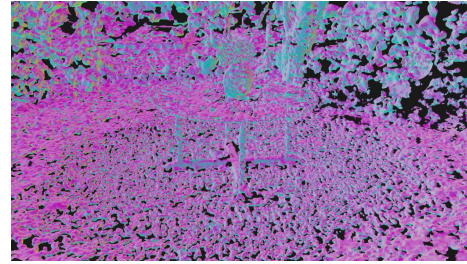
Instead, SuGaR samples surface points at locations where the accumulated opacity reaches a predefined threshold, effectively identifying points close to the true surface (see Fig. 3.3d). These sampled points are then passed to a Poisson surface reconstruction algorithm to obtain mesh output of high quality (see Fig. 3.3e).



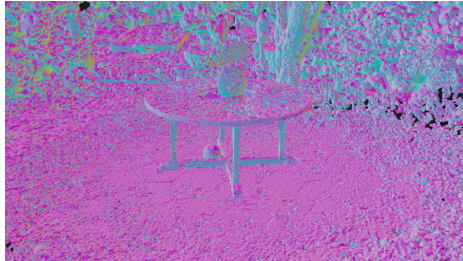
(a) Ground truth image of the scene



(b) M. Cubes, no regularization



(c) Poisson from Gaussian origins



(d) SuGaR extraction, no regularization



(e) SuGaR extraction, with regularization

Figure 3.3: SuGaR ablation study, demonstrating the importance of surface regularizers and SuGaR mesh extraction pipeline. Images taken from a conference keynote by Antoine Guédon, the author of SuGaR [14].

3.1.4 2D Gaussian Splatting (2DGS)

2DGS [18], much like SuGaR [15], uses flat Gaussian primitives to represent the radiance field. This encourages the Gaussians to form a thin shell around object surfaces rather than filling volumes, leading to improved surface reconstruction. However, 2DGS introduces a key difference by explicitly setting the Gaussian’s third scaling coefficient to zero (see Eq. (2.5)), converting the scene representation into a true 2D Gaussian point cloud.

A significant portion of the 2DGS paper is dedicated to redefining the projection and rasterization process to support these degenerate Gaussians. In 3DGS, flat Gaussians are an edge case that can collapse into a line segment when viewed nearly edge-on, leading to unstable projections, incorrect rasterization, and exploding gradients. 3DGS mitigates this by thresholding or discarding problematic splats. SuGaR partially addresses the issue but also relies on thresholding in edge cases. 2DGS avoids the need for such heuristics by using a stable ray-splat intersection model and affine projection.

A notable distinction from SuGaR is that 2DGS does not discourage overlap between neighboring Gaussians. Instead, splat overlaps are allowed; they smooth out depth variations and promote continuity in the reconstructed surface. This design choice improves consistency but leads to over-smoothing in regions with fine detail.

To facilitate high-fidelity surface reconstruction, 2DGS introduces two regularization terms:

- **Depth Distortion Loss** $\mathcal{L}_d$ minimizes the spread of depths among the contributing Gaussians along a ray. It is defined as:

$$\mathcal{L}_d = \sum_{(i,j)} w_i w_j |z_i - z_j|, \quad (3.3)$$

where w_i and w_j are the compositing weights and z_i, z_j are the depths of contributing splats along a pixel ray.

- **Normal Consistency Loss** $\mathcal{L}_n$ encourages the splat normals to align with the surface normal estimated from the depth map. It is given by:

$$\mathcal{L}_n = \sum_i w_i (1 - \langle \mathbf{n}_i, \mathbf{N}(x_i) \rangle), \quad (3.4)$$

where $\mathbf{n}_i$ is the Gaussian's predicted normal, $\mathbf{N}(x_i)$ is the depth map normal at pixel x_i , and w_i is the compositing weight.

After training, mesh extraction is performed via TSDF fusion. The inputs to TSDF fusion are the dense depth maps and surface normals obtained by rendering the 2D Gaussian point cloud from multiple camera views.

3.1.5 Quadratic Gaussian Splatting (QGS)

Quadratic Gaussian Splatting [64] extends 2DGS by introducing quadric surfaces as support geometry for each Gaussian, rather than planar disks. This enables more accurate modeling of curved surfaces and leads to improved geometric fidelity. In this sense, QGS generalizes 2DGS by embedding each Gaussian on a locally curved surface patch.

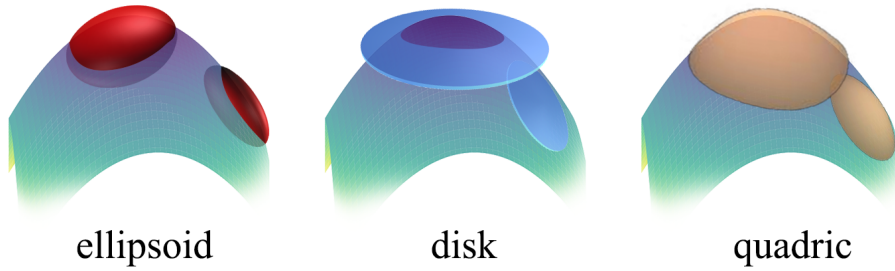


Figure 3.4: Surface fitting comparison. Quadrics capture curvatures with a higher level of approximation. Image from Zhang et al. [64].

- **Quadric surface.** A quadric surface in homogeneous coordinates is defined as the set of points

$$f(x, y, z) = \mathbf{x}^T Q \mathbf{x} = 0, \quad \text{where} \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix},$$

and $Q \in \mathbb{R}^{4 \times 4}$ is a symmetric matrix of the form:

$$Q = \begin{bmatrix} A & B & C & D \\ B & E & F & G \\ C & F & H & I \\ D & G & I & J \end{bmatrix}.$$

The symmetry of Q ensures that the quadratic form $\mathbf{x}^T Q \mathbf{x}$ yields a real-valued scalar. Hence, the quadric defines a scalar-valued function

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}.$$

- **Quadric surface primitive.** QGS assigns a quadric surface to each 3D Gaussian, derived implicitly from the Gaussian’s covariance matrix. The eigenvalues and eigenvectors of the covariance define the local curvature and orientation of the surface. In practice, the resulting shapes are anisotropic convex paraboloids with guaranteed smoothness and convexity due to the positive semi-definiteness of the covariance matrix. This regularity ensures that geodesic distances on the surface are well-defined and numerically stable.

During rendering, QGS evaluates a Gaussian’s opacity contribution by projecting the camera ray onto the Gaussian’s local quadric surface. The geodesic distance from the Gaussian center to the ray–surface intersection point is computed, and used as input to a Gaussian kernel to determine the opacity.

- **Regularization and implementation details.** QGS adopts and extends both regularizers from 2DGS (see Eq. (3.3), Eq. (3.4)). The normal consistency loss is enhanced using a curvature-based weighting scheme, which improves robustness near edges and surface discontinuities. Furthermore, QGS integrates the stricter depth-sorting strategy of StopThePop [41] to reduce banding artifacts during mesh extraction.

Following 2DGS, QGS performs mesh reconstruction via truncated signed distance fusion (TSDF) using depth maps rendered from the trained Gaussian representation. For visual comparison of the two methods, refer to Figure 3.5.

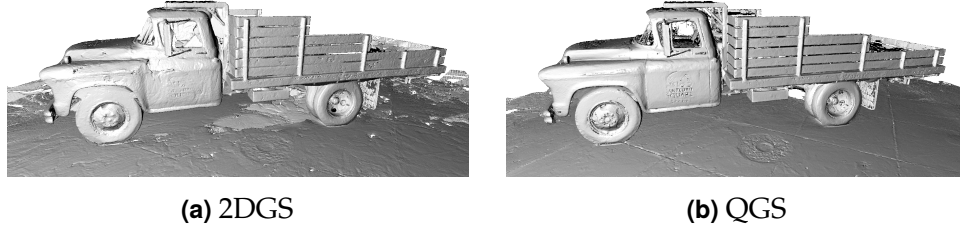


Figure 3.5: Qualitative comparison of meshes extracted by QGS and the baseline 2DGS method. The depicted scene belongs to the Tanks and Temples dataset [27]. Images taken from QGS [64].

3.1.6 PGSR

PGSR extends the 3D Gaussian Splatting (3DGS) framework [22] by introducing a surface-aware rendering pipeline tailored for high-fidelity surface reconstruction. As in 3DGS, the scene is represented using a set of 3D Gaussian primitives optimized through photometric supervision.

The core innovation of PGSR lies in its method for generating accurate depth and normal maps by rendering flattened Gaussian representations. These maps are subsequently fused into a mesh using the TSDF fusion approach. However, rendering improvements alone are insufficient for high-quality reconstruction. The Gaussians trained purely under photometric loss, as in standard 3DGS, are not explicitly encouraged to align with the scene’s underlying surface geometry. To address this, PGSR introduces geometric regularization during training, including both single-view and multi-view consistency losses, which guide the Gaussians toward surface-consistent configurations.

Unbiased depth rendering of PGSR is made possible by approximating each Gaussian as a flat disk. This is achieved by minimizing its smallest scale factor:

$$L_s = ||\min(s_1, s_2, s_3)||_1.$$

Cases where multiple scale coefficients are nearly equal are resolved by selecting the normal direction such that the angle between the viewing direction and the normal exceeds 90 degrees. This ensures that the surface is viewed from the front side.

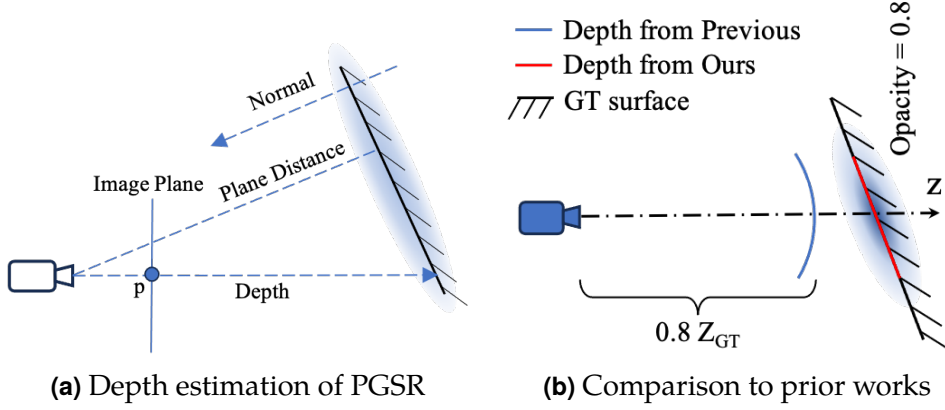


Figure 3.6: Unbiased depth estimation of PGSR, demonstrated on a single Gaussian primitive. Depth is estimated as the distance of Gaussian tangent plane along the camera ray, rather than the distance of the camera from Gaussian origin. Images adapted from Chen et al. [5].

From an arbitrary viewpoint, PGSR renders a normal map via alpha blending of the normals of visible flattened Gaussians. A preliminary depth map is computed analogously by alpha blending the distances from the camera center to each Gaussian’s tangent plane, measured along the normal. Unlike prior approaches that interpret this scalar as the true depth, PGSR computes the intersection point of the viewing ray with the rendered tangent plane (see Fig. 3.6). This produces an unbiased depth map that better reflects the underlying surface geometry.

Single-view regularization: PGSR enforces local geometric consistency by assuming that each pixel and its immediate neighbors lie on an approximately planar surface. For each pixel, the depth values of its neighbors are used to estimate a local plane normal. A regularization term penalizes deviations between these estimated normals and the rendered normal map. To reduce the influence of depth discontinuities, the regularization term is weighted by a function of the image gradient, thereby attenuating the constraint in regions with strong intensity edges.

Multi-view geometric regularization further improves robustness to noise, blur, and lighting artifacts in ground truth images by enforce-

ing consistency across multiple views. Given a pair of neighboring camera poses, PGSR transforms each pixel in the reference view to the neighboring view using a homography derived from the rendered depth and normal maps. A reverse transformation is then applied to project the pixel back to the reference view. The forward-backward projection error is minimized as a regularization term, encouraging the consistency of geometric predictions across views.

Additional enhancements: PGSR also incorporates a multi-view photometric consistency loss, inspired by classical multi-view stereo (MVS) techniques. To handle variations in exposure across images, PGSR learns per-image exposure coefficients, allowing it to adjust the brightness of the rendered image before computing photometric loss. If the structural similarity (SSIM) between the rendered image and ground truth is low, the exposure-adjusted version is used to compute the loss, improving robustness to illumination differences.

3.1.7 Gaussian Opacity Fields (GOF)

The Gaussian Opacity Fields (GOF) method [63] addresses the same fundamental limitation identified in SuGaR (see Section 3.1.3): the density field produced by a fully trained 3D Gaussian point cloud is often too sparse to support reliable surface reconstruction. While many Gaussian-based approaches circumvent this issue by adopting 2D Gaussian primitives, GOF tackles the problem directly by introducing a formulation that yields a volumetric density field of sufficient fidelity for mesh extraction.

The GOF method begins by considering a ray defined as

$$x(t) = o + tr,$$

where $o \in \mathbb{R}^3$ is the camera center, $r \in \mathbb{R}^3$ is the normalized ray direction, and $t \in \mathbb{R}^+$ denotes the depth along the ray. Given a 3D Gaussian $\mathcal{G}_k$, the contribution of $\mathcal{G}_k$ to a point along the ray is computed by projecting $\mathcal{G}_k$ onto the 1D domain of the ray. The resulting distribution along t is a 1D Gaussian. This formulation enables the evaluation of volumetric density at any arbitrary point $x(t)$ along the ray.

As the ray traverses the scene, the opacities contributed by each Gaussian are integrated along t using an alpha compositing model analogous to NeRF, yielding the *accumulated opacity* $O(o, r, t)$. This

3. THEORY

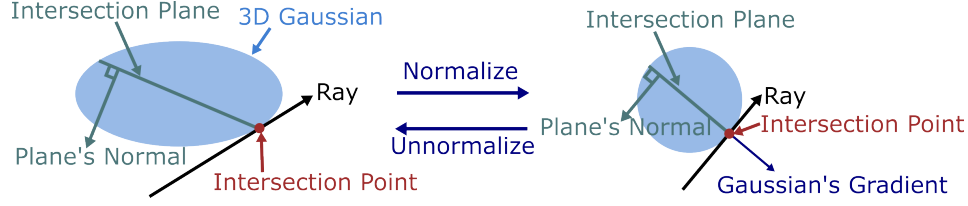


Figure 3.7: Approximation of Gaussian’s normal as the normal of the ray-Gaussian intersection plane. First, ray is transformed into the Gaussian coordinate system and normalized using the Gaussian’s scales. In this normalized coordinate system, the ray-Gaussian intersection plane is perpendicular to the ray, making the normal the inverse of the ray direction. The intersection plane is then transformed back to world space. Figure adopted from GOF [63].

accumulated opacity captures how much the ray has been occluded up to depth t .

Since a 3D point x might be visible from multiple training views or ray directions, GOF defines the final opacity of a point x as the minimal accumulated opacity among all rays passing through x :

$$O(x) = \min_{(o,r)} O(o, r, t).$$

Taking the minimum ensures that if a point is transparent from any view, it is treated as transparent in the final field, providing a conservative estimate of object solidity.

The collection of opacities $O(x)$ for all points in the scene constitutes the **Gaussian Opacity Field**.

In addition to the concept of Gaussian Opacity Fields, GOF introduces changes to the training pipeline to produce a Gaussian point cloud better suited for surface reconstruction. Specifically, the method incorporates the depth distortion (see Eq. (3.3)) and normal consistency (see Eq. (3.4)) regularizers from 2DGS [18]. Unlike 2DGS, however, the Gaussians in GOF do not possess explicit surface normals. To enable the use of these regularizers, GOF estimates the normal of each Gaussian as the normal of the intersection plane defined by the viewing ray direction (see Fig. 3.7).

For mesh extraction, GOF constructs a spatially adaptive tetrahedral grid that allocates finer resolution to regions densely populated

with Gaussian primitives. Rather than using the marching cubes algorithm, GOF employs marching tetrahedra: surface extraction is performed by evaluating the opacity at the vertices of each tetrahedron. To more precisely locate surface intersections, binary search is applied along tetrahedral edges instead of relying on linear interpolation, which accounts for the non-linear behavior of the opacity field. A qualitative comparison to surface reconstruction methods discussed in previous subsections can be seen in Figure 3.8.

This extraction strategy offers improved adaptability to local surface structure. The tetrahedral grid allows for large cells in regions with minimal changes in opacity, while densely populating areas with high variation. As a result, the extracted mesh is not constrained to a uniform voxel resolution, as is the case with marching cubes, and achieves a more efficient distribution of triangles.

3. THEORY



Figure 3.8: Qualitative comparison of meshes produced by surface reconstruction methods based on Gaussian radiance fields. Surface reconstruction performed on scenes of the Mip-NeRF 360 dataset [2]. Images taken from PGSR [5].

3.2 Reconstruction of Transparent Surfaces

The reconstruction of transparent surfaces remains an open problem, both within and beyond the context of radiance fields. While radiance field methods have demonstrated strong performance in view synthesis, including in scenes containing transparent objects, their application to surface reconstruction in such cases is still limited.

Only a small number of radiance field-based methods explicitly target transparent materials. Among these, most focus on depth estimation for robotic grasping rather than on recovering full surface geometry. This section outlines the core concepts of existing approaches.

While the depth maps produced by methods invented for the purpose of robotic grasping are typically insufficient for accurate surface reconstruction, these methods introduce influential ideas that merit their inclusion.

3.2.1 Dex-NeRF

Dex-NeRF [19] enhances NeRF’s depth estimation for robotic grasping of transparent objects by addressing two challenges specific to transparent surfaces (see Fig. 3.9).

First, Dex-NeRF identifies glossiness, and the resulting specular highlights, as a common trait of transparent objects. These highlights are particularly informative, as they appear opaque from certain viewpoints while remaining transparent from others. Standard NeRF models tend to assign partial opacity to such regions, rather than treating them as empty space. To increase the presence of specular cues during training, Dex-NeRF captures ground truth images under a 5×5 grid of light sources instead of a single source.

Second, the partial opacity used by NeRF to represent transparent objects results in densities that are insufficient to terminate rays reliably. Rather than relying on expected termination depth (see Section 3.1.1), Dex-NeRF terminates a ray when its transmittance drops below a predefined threshold.

3. THEORY

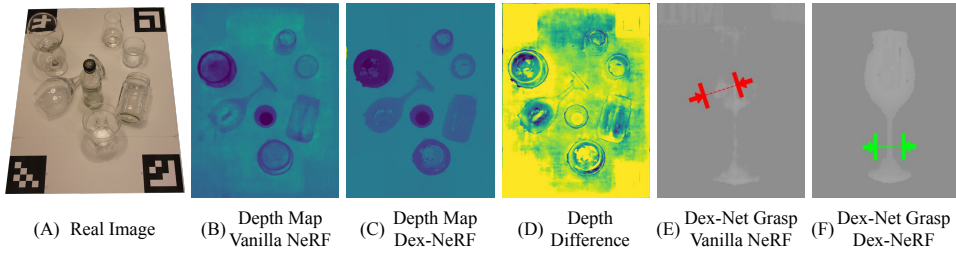


Figure 3.9: Dex-NeRF uses transparency-aware depth rendering to render depth maps that can be used for grasp planning. In contrast, Vanilla-NeRF’s depth maps are filled with holes and result in poor grasp predictions. Figure adopted from Dex-NeRF [19].

3.2.2 Residual-NeRF

Residual-NeRF [11] is designed for robotic grasping tasks involving transparent objects. It assumes a static scene that includes a robotic arm and other opaque objects, but no transparent ones. First, a NeRF is trained to model this baseline scene. A second NeRF is then trained on the same scene after transparent objects have been added. By comparing the depth maps predicted by the two NeRFs, the method estimates the shape and pose of the added transparent objects.

The use of residuals (computing changes relative to a baseline) is a common general strategy not limited to implicit representations. Residual-NeRF exemplifies this approach within the NeRF framework. A corresponding example in the Gaussian Splatting domain is Clear-Splatting [1].

3.2.3 GraspNeRF

GraspNeRF [10] reconstructs transparent objects for robotic grasping by combining neural rendering with supervised TSDF prediction. A NeRF is first trained to model the scene from multi-view images. Then, a pre-trained TSDF predictor takes NeRF’s volumetric features on a 3D grid as input and outputs a dense TSDF field. This representation is used directly for grasp planning, bypassing explicit surface extraction or depth rendering. Similarly to aforementioned methods, however, the quality of the resulting representation is insufficient for meaningful mesh extraction.

3.2.4 Ref-NeRF

Ref-NeRF [52] significantly enhances NeRF’s capability to model view-dependent effects such as specular reflections, particularly in glossy and transparent materials. The method introduces several key innovations:

Recall the geometry and appearance networks introduced in NeRF (subsection 2.2.1). Ref-NeRF extends this formulation by decomposing the emitted color into a view-independent diffuse component and a view-dependent specular component, assigning the diffuse prediction to the geometry network due to its invariance to viewing direction.

Rather than training the appearance network to predict color as a function of the viewing direction ϕ_o , Ref-NeRF trains it as a function of the viewing vector reflected about the normal of the intersected surface ϕ_r . This change reflects the fact that specularity is a property of the surface, rather than a property of the camera. Figure 3.10 demonstrates that the reparametrized function of ϕ_r is smoother than the function of viewing direction ϕ_o , allowing the appearance network to approximate it more accurately.

Surface normals are estimated in standard NeRF using gradients of the predicted density field, but these estimates tend to be noisy. Ref-NeRF instead modifies the spatial MLP to explicitly predict vectors which can be used to calculate normals alongside density and diffuse color. These vectors are not only used during inference, but also supervised during training, leading to smoother and more accurate normal predictions. Additionally, the spatial MLP predicts the material property of roughness ρ .

Standard NeRFs often incorrectly model specular highlights as volumetric emitters behind partially transparent surfaces. To address this, Ref-NeRF introduces a normal orientation regularizer that penalizes samples contributing to rendering from behind a surface boundary.

$$\mathcal{R}_o = \sum_i w_i \left(\max \left(0, \tilde{\mathbf{n}}_i \cdot \hat{\mathbf{d}} \right) \right)^2, \quad (3.5)$$

where:

- $\mathcal{R}_o$: Normal orientation regularization loss,
- w_i : Termination weight for sample i ,

3. THEORY

- $\hat{\mathbf{n}}_i$: Gradient-based normal at sample i ,
- $\hat{\mathbf{d}}$: Viewing direction (unit vector).

This regularization works because the normal vector is defined as the normalized gradient of the density field. A positive dot product with the viewing direction indicates that the density is decreasing along the ray, which typically occurs behind the surface. The loss thus penalizes samples that contribute strongly to rendering while lying beyond a surface boundary.

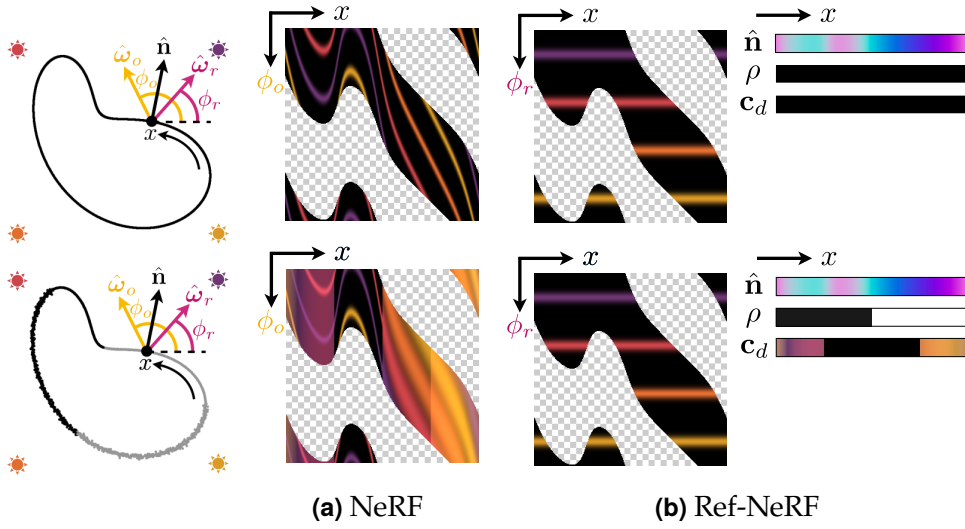


Figure 3.10: Position-angle visualization of emitted radiance in NeRF and the reparametrized Ref-NeRF. Top: changing specular highlights along reflective object. Bottom: changing materials along rough object. Images adopted from Verbin et al. [52].

3.2.5 α Surf

α Surf uses an implicit representation implemented using an explicit voxel grid, first shown in Plenoxels [12]. The representation blurs the border of NeRF and GS paradigms in an interesting way.

Adapting the scene representation of Plenoxels, α Surf encodes NeRF volumetric properties (density and view-dependent color) as

continuous fields. Unlike Plenoxels, it explicitly separates visual opacity from physical occupancy, addressing their conflation in scenes with transparent objects. The scalar field $\mathbb{R}^3 \rightarrow \mathbb{R}$ representing physical density is denoted by $\delta(x)$.

Each volumetric property is stored at the vertices of an explicit voxel grid, and values at arbitrary points in space are obtained via trilinear interpolation of the eight surrounding grid vertices.

The scalar field $\delta(x)$ in α Surf defines level sets but does not satisfy the Eikonal constraint¹ to be considered an SDF. As a result, while the field indicates whether a point lies below, on, or above a level set, its value does not represent the distance from the surface. This design choice allows the use of a surface variance regularizer, which encourages the formation of more geometrically accurate level sets.

In contrast to all previously discussed SDF methods, α Surf defines a surface as the union of a finite discrete number of level sets, rather than a single one. Formally, for a hyperparameter n , the set of level values $\tau = \{\tau_j\}_{j=1}^n$ defines the surface as the union of corresponding level sets:

$$\mathcal{S} = \left\{ x \in \mathbb{R}^3 \mid \exists \tau \in \tau : \delta(x) = \tau \right\}.$$

α Surf is trained using a standard photometric loss augmented with custom regularizers. Scene images are rendered via a raymarching procedure. For each camera ray, the algorithm first determines which voxels in the explicit grid the ray intersects, in an operation known as ray-voxel traversal. Within each intersected voxel, the ray equation is substituted into the trilinear interpolation of the scalar field, resulting in a cubic polynomial. Solving this polynomial for each level set value τ_j yields candidate ray-surface intersection points. Complex roots and those outside the bounds of the voxel are discarded. The remaining intersections are sorted by depth and composited using alpha blending to produce differentiable pixel values.

This highlights the role of having multiple level set values τ_j . Each level set contributes to the alpha compositing process, ensuring that even thin geometry, where some level sets may be absent, is still cap-

1. The Eikonal constraint requires that the gradient of a signed distance field has unit norm: $|\nabla f(x)| = 1$ for all $x \in \mathbb{R}^3$.

3. THEORY

tured accurately. In such cases, lower-valued level sets can compensate by contributing sufficient visual opacity to saturate the alpha blending.

To improve surface quality, α Surf incorporates several regularization terms into the loss:

- $L_C(r)$ encourages adjacent level sets to remain close, reducing ambiguity between surface layers.
- L_{n1} , L_{n2} , and L_{ffi} promote surface smoothness.
- L_H is an entropy-based regularizer on the opacity field that encourages binary transparency values, similarly to opacity regularization of SuGaR (Section 3.1.3).
- L_α , called the sparsity loss, penalizes unnecessary opacity.

The role of these regularizers is not only to improve surface reconstruction quality. The L_H and L_α regularizers are designed to incentivize the representation to distinguish between the visual opacity and physical density.

For surface evaluation, α Surf extracts point clouds rather than meshes. This is done by rotating the scene around a central point and densely emitting rays in a spiral pattern. Points are placed at the alpha-weighted ray–surface intersections obtained through the procedure described above. Notably, α Surf does not perform mesh extraction. Classical methods such as marching cubes are incompatible with its multi-level-set representation, as they assume a single iso-surface. Although α Surf could potentially be extended to mesh reconstruction via TSDF fusion or Poisson reconstruction as was done in SuGaR (Section 3.1.3), the authors do not explore this direction. Figure 3.11 provides a qualitative evaluation of α Surf and a comparison to the general-purpose surface reconstruction method of NeuS [54] discussed in Section 3.1.2.

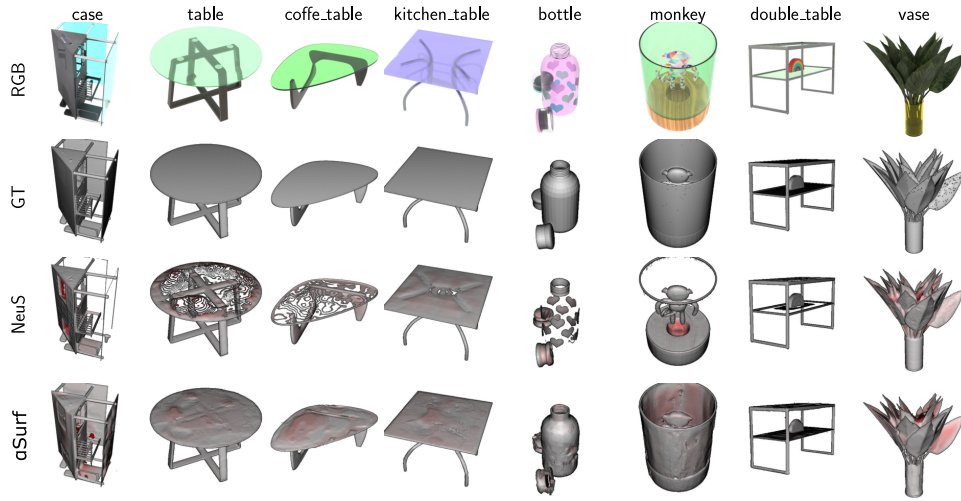


Figure 3.11: α Surf surface reconstruction performance qualitatively evaluated on a simple synthetic dataset. The images depict a comparison of extracted point clouds to the generic surface reconstruction method NeuS [54] and ground truth data. Figure adapted from Wu et al. [57].

3.2.6 Car-GS

Car-GS [30] is a 2DGS-based method developed specifically for surface reconstruction of cars. It introduces two key innovations to improve the handling of reflective and transparent surfaces:

- **View-dependent 2D Gaussian primitives (VDGs)** are introduced for each training image to model specular reflections. These image-space Gaussians are discarded at inference time, enabling the core 3D representation to generalize across viewpoints.

The use of VDGs to capture and discard specular reflections may appear to contradict the findings of Dex-NeRF [19], which treats specularities as informative cues (discussed in Section 3.2.1). However, specular reflections behave differently in Gaussian point cloud training compared to NeRF. Since each Gaussian primitive learns view-dependent appearance and its position is optimized during training, a primitive may initially capture correct specular behavior in one region, only to be relocated to another where such reflections are physically incorrect.

3. THEORY

- **Dual-opacity modeling** replaces the single opacity field used in standard Gaussian Splatting. One opacity field governs visual appearance (RGB rendering), while a second is used for geometry (depth and normals), more accurately reflecting ray behavior in transparent or refractive regions.

Although Car-GS proposes novel extensions, its applicability is limited. The method is tailored specifically to cars and depends on two pretrained models: one for semantic segmentation and another for surface normal estimation. It has not yet been peer-reviewed or formally published, and the absence of public code restricts its reproducibility and broader adoption.

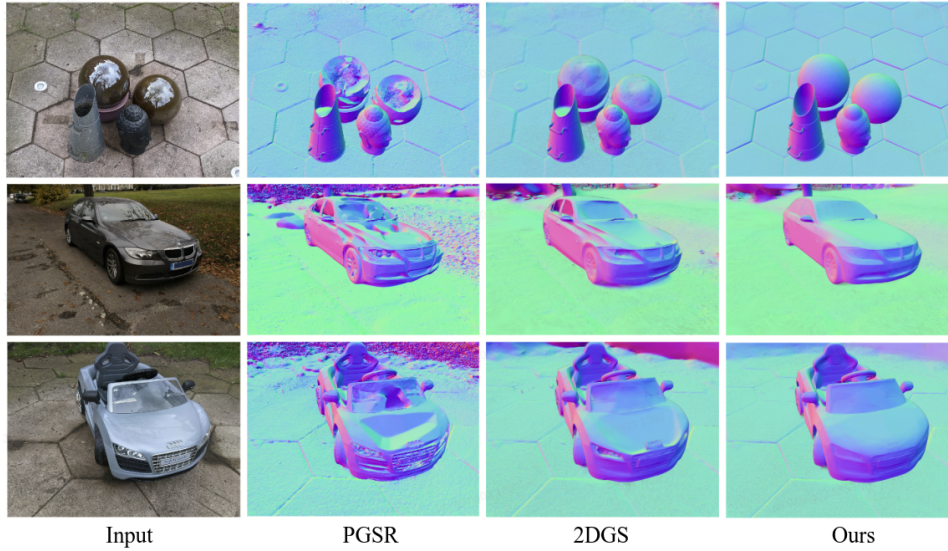


Figure 3.12: Qualitative comparison of Car-GS [30] to other surface reconstruction methods based on Gaussian radiance fields. Figure adopted from Car-GS [30].

3.2.7 TSGS

Transparent Surface Gaussian Splatting (TSGS) [31] is a very recent method designed specifically for reconstruction of transparent surfaces using the explicit representation of Gaussian point clouds. They demonstrate the effectiveness of their approach on their own synthetic dataset called TransLab. The dataset contains models of laboratory glassware.

The TSGS training pipeline is conceptually complex, combining multiple pretrained neural networks with a staged optimization process. At a high level, the method relies on heavy use of external networks to predict surface normals, de-lighted images (images in which specular reflections have been removed), and segmentation masks of transparent objects from ground truth input images. These predicted maps serve as priors in regularization terms during the subsequent training of the 3D Gaussian point cloud.

The use of a de-lighting network to remove specular reflections from ground truth images is an interesting alternative to VDGs used in Car-GS (see Sec. 3.2.6) for the same purpose.

Training is divided into two stages. The first stage focuses on geometry and uses the de-lighted images along with normal and segmentation priors. During this phase, each Gaussian primitive is augmented with a learnable attribute $\tau \in [0, 1]$, representing the degree to which it belongs to a transparent surface. After the first stage, τ is fixed.

The second stage refines appearance. It replaces spherical harmonics with a more accurate system of anisotropic spherical Gaussians (first done in Spec-Gaussian [60]) to better capture view-dependent effects and employs an additional neural network to predict specular reflection colors. Training at this stage is performed on the original (non-de-lighted) ground truth images.

For surface extraction, TSGS adopts a multi-view depth fusion approach similar to PGSR, estimating depth maps from multiple viewpoints and aggregating them to produce a unified surface representation. The depth maps are obtained using a first-surface depth estimation strategy specifically adapted to Gaussian Splatting. The standard 3DGS computes depth as an opacity-weighted average of Gaussian depths via alpha blending. When calculating the depths of transparent surfaces, this approach fails and leads to what TSGS calls

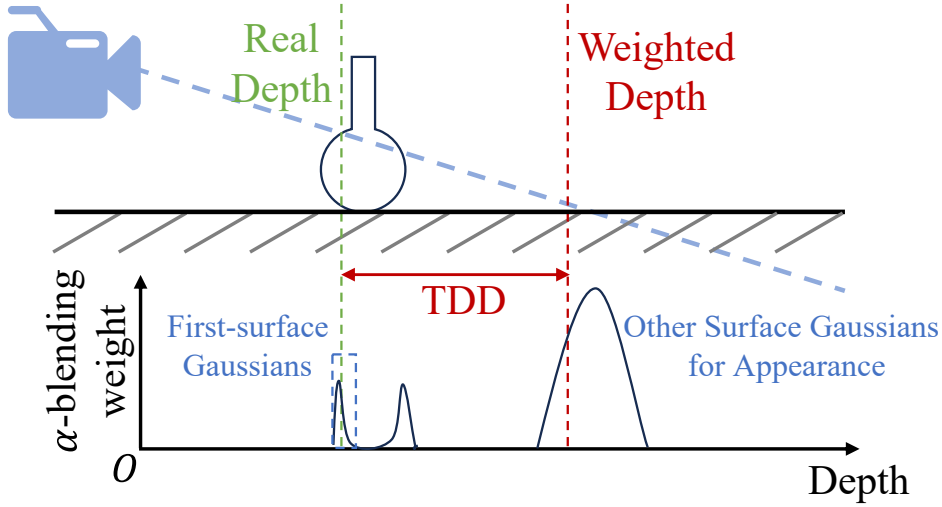


Figure 3.13: Transparency-depth dilemma. Scene background contributes to depth estimation due to low opacity of Gaussians representing the transparent object. Image taken from TSGS [31].

the *transparency-depth dilemma* (as shown in Fig. 3.13). TSGS instead distinguishes transparent regions using the learned τ parameter and performs a windowed search along the viewing direction to locate the point of greatest transmittance drop. This point, corresponding to the first significant accumulation of opacity, is treated as the first surface intersection and is used for mesh extraction.

Although the term “transparency-depth dilemma,” introduced by TSGS, is conceptually applicable to other radiance field representations, it is not yet established in the literature. Moreover, TSGS defines and applies it specifically in the context of 3D Gaussian point clouds. For these reasons, we use the term only in the context of TSGS, rather than adopting it as a general concept throughout the thesis.

A qualitative comparison of TSGS to other surface reconstruction methods can be seen in Figure 3.14.

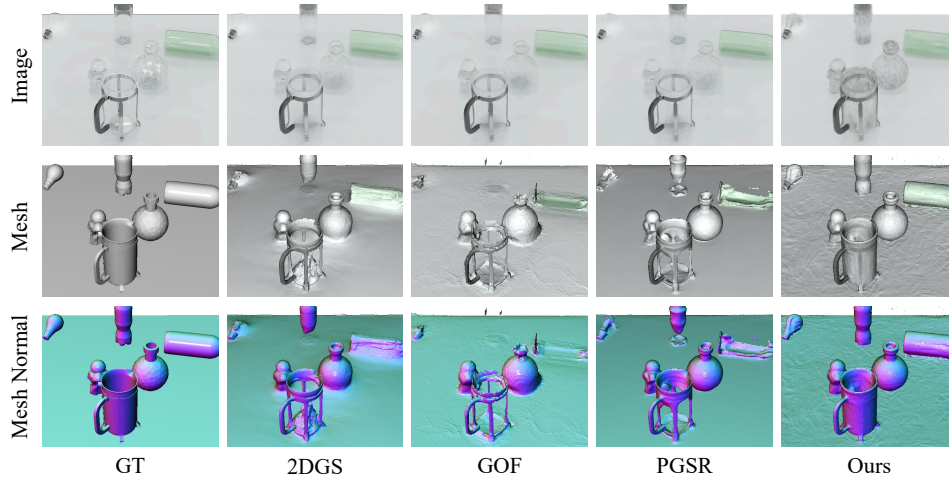


Figure 3.14: Qualitative comparison of TSGS to general purpose surface reconstruction techniques based on radiance fields, performed on the TSGS synthetic TransLab dataset. Figure adopted from TSGS [31].

3.3 Chapter Summary

The theoretical part of this thesis presented a range of radiance field-based methods and examined their respective approaches to surface reconstruction. Their strengths, limitations, and potential extensions were described and compared, with particular attention given to the unique challenges associated with reconstructing transparent surfaces.

Across recent methods explicitly designed for reconstructing transparent surfaces, there is a consensus that the primary limitation of general-purpose surface reconstruction techniques in such scenes lies in their conflation of visual opacity and physical occupancy. While this assumption holds in the case of opaque surfaces, it becomes fundamentally invalid in the presence of transparency. Decoupling these two properties introduces new challenges during training, particularly concerning the supervision and regularization of the additional degree of freedom introduced.

In this context, implicit representations exhibit a distinct advantage. Early approaches for depth estimation have demonstrated the ability to leverage specular highlights as informative training cues. In contrast, explicit methods typically treat such highlights as undesirable artifacts,

3. THEORY

aiming to detect and remove them rather than use them constructively. To compensate, they rely on priors derived from pre-trained deep learning models.

Another common issue across both paradigms is that, even when transparent geometry is convincingly captured and renders well in novel view synthesis, extracting a clean surface remains problematic. In NeRF-based methods, for instance, transparent surfaces tend to correspond to regions of low density. Attempting to extract lower-density level sets often introduces spurious floating artifacts in other regions of the scene, whose surfaces correspond to higher-density levels.

This issue is addressed in a novel manner by the hybrid implicit-explicit method α Surf, which represents the scene as a union of multiple level sets of a scalar geometry field. By modulating these surfaces with an independently learned opacity field and applying alpha compositing during rendering, α Surf enables the reconstruction of thin and transparent objects while mitigating the presence of artifacts in opaque regions.

Despite these advances, the surveyed methods fall short of providing a practically effective solution to transparent surface reconstruction. NeRF-based methods remain computationally intensive, insufficiently accurate, and challenging to extend with newer training techniques. α Surf, while innovative, inherits significant limitations from its voxel grid representation. Meanwhile, GS-based approaches depend heavily on priors from pre-trained networks, which compromises generalizability and introduces pipeline complexity with limited theoretical justification.

Furthermore, surface reconstruction methods for transparent objects are typically developed on top of a chosen novel view synthesis framework but often introduce substantial modifications to the training pipeline and scene representation. As a result, they diverge significantly from their original foundations, which can make it difficult to incorporate future advances from the novel view synthesis domain.

4 Method

The reconstruction of transparent surfaces poses a unique challenge not present in the reconstruction of opaque geometry: visual opacity can no longer be treated as a direct proxy for physical density. Instead, these two properties must be modeled as separate variables. This separation, however, introduces an additional degree of freedom into the scene representation, which can be easily misused during training if not properly constrained. Existing methods address this issue either by introducing explicit regularization terms [57], or by leveraging geometric priors obtained from pre-trained neural networks [30, 31].

As a result, surface reconstruction methods tailored for transparent objects often deviate significantly from their general-purpose counterparts. This limits their compatibility with ongoing advancements in radiance field training.

4.1 Motivation and Overview

The method proposed in this thesis aims to extend a generic surface reconstruction technique to better handle transparency, while preserving its original structure and training process. Rather than introducing an explicit physical density field into the training pipeline or modifying the training objective with additional regularizers, it relies solely on the visual opacity learned during standard training. After training, a segmentation method is used to aggregate Gaussians of transparent objects from multiple intermediate iterations of optimization. Their visual opacities are then increased to match the opacities of opaque objects present in the scene. The adjusted representation is used for mesh extraction instead of the trained one.

The simple change greatly increases surface reconstruction quality for transparent objects, as this chapter demonstrates in a qualitative evaluation on the "Counter" scene from the Mip-NeRF 360 dataset [2].

The superior results indicate that:

- Sampling across training iterations is an underexplored strategy and a potential alternative to the use of geometric priors for training guidance.

- Additional refinement of radiance fields after training can significantly improve the quality of subsequent mesh extraction.
- The existing SAM-based segmentation techniques can be used to accurately target Gaussian primitives of transparent surfaces.

4.2 Baseline method analysis

For the practical part of this thesis, the method of Gaussian Opacity Fields (GOF) [63] (see Section 3.1.7 for more detail) was chosen as the baseline to build upon. Although GOF is no longer the state-of-the-art in surface reconstruction from radiance fields, it offers a number of structural advantages over more recent alternatives. In addition to providing an adaptive mesh extraction pipeline, GOF retains the original 3D Gaussian representation used in 3D Gaussian Splatting (3DGS) without severely altering the data structure or rendering pipeline. Its only major change to the 3DGS framework is the introduction of surface regularizers commonly present in other surface reconstruction methods [18]. The GOF pipeline consists of four main stages:

1. A 3D Gaussian point cloud is trained to model the scene.
2. The Gaussian opacity field is established as a continuous scalar field derived from the visual opacity of the Gaussian primitives.
3. An adaptive tetrahedral grid is established over the Gaussian opacity field.
4. Mesh extraction is performed by the Marching Tetrahedra mesh extraction algorithm and binary search over the tetrahedral grid to accurately locate opacity level sets.

Firstly, we replicate the surface reconstruction experiment conducted in the original GOF study, using the Mip-NeRF 360 dataset [2]. Our experimental results confirm that GOF produces high quality meshes of the scenes, but fails to reconstruct transparent objects (as shown in Fig. A.1b). The issue persists when extracting the lower opacity level-set of 0.1 (see Fig. A.1c).

A closer inspection of the trained Gaussian point cloud supports the hypothesis that their training stage suffers from the conflation of

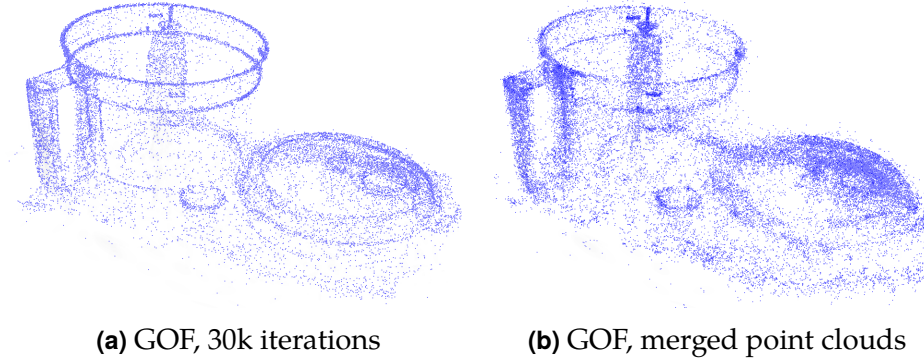


Figure 4.1: Origins of Gaussian primitives representing a transparent object. Both trained by GOF. Left: unmodified representation trained for full 30,000 iterations. Right: Merged representation of 1k, 2k, 3k, 4k iteration checkpoints.

visual opacity and physical density. The training produces sparsely placed Gaussians (see Fig. 4.1a) of low opacities as the representation of transparent surfaces. The opacity contributions of such primitives are insufficient for surface reconstruction.

The GOF pipeline allows for the introduction of an intermediate step between the first and second stage. By populating the fully trained Gaussian point cloud with additional Gaussian primitives (see Fig. 4.1b) and directly increasing the opacity of Gaussian primitives belonging to transparent objects, our method obtains a physically meaningful representation better suited for surface reconstruction. The representation is processed by the remaining stages of the GOF surface reconstruction pipeline.

4.3 Complete Method

Post-training representation adjustment is carried out in a manner which only affects the Gaussian primitives of transparent objects:

1. A snapshot of the training process is captured at multiple iterations: 1,000, 2,000, 3,000, 4,000, 30,000.
2. Transparent objects are segmented separately for each of the training iterations using the SAGA [3] method supplied by manual one-shot click prompts (as shown in Fig. 4.2).
3. The opacities of Gaussian primitives belonging to transparent objects are set to match the maximum opacity present in the scene.¹
4. The Gaussian point clouds are merged into a single representation. Opacity-enhanced Gaussians of all iterations are preserved. All other Gaussian primitives of iterations 1k, 2k, 3k, 4k are discarded. The unmodified Gaussian primitives of the final training iteration 30,000 are preserved and serve as scene background.

Sampling across iterations is meaningful under the assumption that after a sufficiently high number of steps, Gaussians lie in close proximity to the surface and can serve as its approximation. We apply the assumption because in the process of training the scene representation, Gaussian primitives shift their positions by small increments in the search of local minima. The full training process takes 30,000 iterations. Since the features of 3D Gaussians need to be mature enough to approximate the surface reliably, yet are most volatile in early training stages, we choose iterations 1k, 2k, 3k, and 4k for sampling.

Segmentation of Gaussian primitives is done using SAGA [3], a SAM-based [26] method that requires manual input in the form of point clicks. Each of the training iterations used for surface reconstruction by

1. The GOF reconstruction pipeline attenuates the opacities of all primitives using the sigmoid activation function. For practical purposes, opacity increases do not matter once sufficiently high.



Figure 4.2: Segmentation process using SAGA.

our pipeline required a separate set of manual prompts. While not practical for large-scale or automated experimentation, SAGA was chosen because it serves as a proxy for emerging open-vocabulary segmentation methods that also build on the Segment Anything Model (SAM). At the time of writing this thesis, numerous promising works have recently been published on open vocabulary segmentation [33, 6, 58] of 3D Gaussians, potentially allowing the segmentation of transparent objects based on a pre-defined set of prompts. All of the aforementioned methods have yet to publish their implementations. However, since these approaches share the same underlying segmentation model, their performance is expected to be similar, regardless of whether they are guided by clicks or language prompts. As open-vocabulary methods mature and become available, SAGA can be replaced by an open-vocabulary method with prompts targeting transparent objects.

The increase of opacity for Gaussian primitives based on segmentation resolves the issue of conflated opacity and physical density. By making the Gaussians appear as an opaque surface, we can rely on the assumption that opacity and density are the same trait. However, the change of opacity alone is not sufficient for surface reconstruction and

4. METHOD

must be accompanied by sampling multiple iterations of the training pipeline, as shown in the ablation Figure A.1d.

4.4 Evaluation and Discussion

The "Counter" scene of the Mip-NeRF 360 dataset [2] was chosen for the evaluation of our method. The scene is frequently used for evaluation of novel view synthesis and surface reconstruction methods. It contains multiple transparent household objects, making it a challenging and representative test case for this work. We provide a qualitative evaluation of our method, using image comparisons of the scene reconstruction performed by the original GOF and our altered pipeline.

Figure 4.3 depicts extracted mesh of the entire reconstructed scene. Figure 4.4 shows normals of the extracted mesh from a different angle. Both figures demonstrate that our method improves the mesh extraction quality of transparent objects, while preserving the same level of quality as GOF in the remainder of the scene.

Our results can be best observed in the side by side comparison of transparent objects cut from the reconstructed mesh 4.5.

Ground truth geometry is not provided in the chosen dataset, and as a result, no quantitative surface reconstruction metrics (e.g., Chamfer distance, Hausdorff distance) are reported.

Our experiments were carried out on an NVIDIA A100 80GB VRAM GPU. SAGA models were trained for 5k iterations with default settings. Mesh cutouts shown in Fig. 4.5 were obtained by removing all vertices outside manually established bounding boxes which span areas of interest.

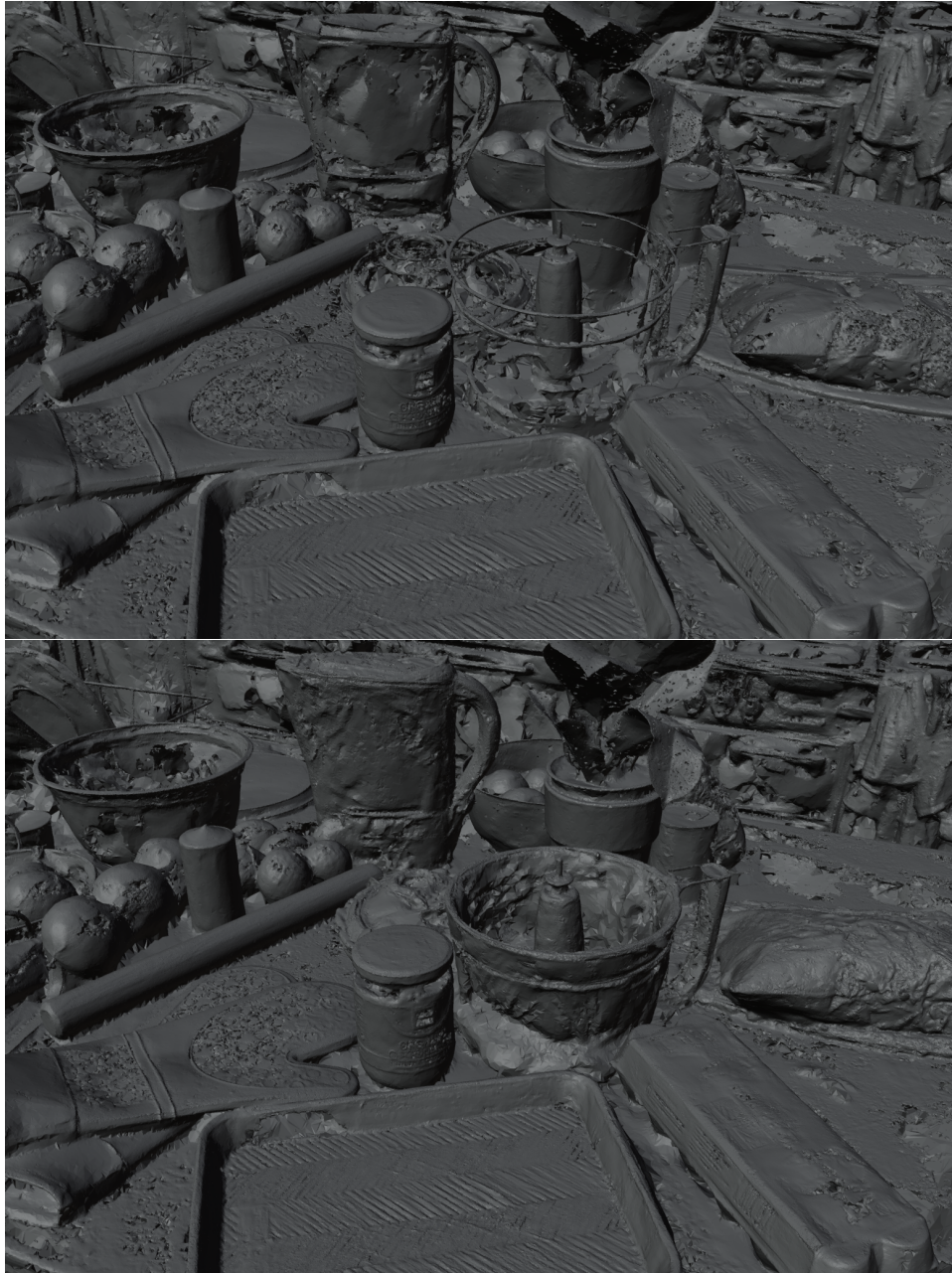


Figure 4.3: The full reconstructed mesh. Top: GOF. Bottom: Ours

4. METHOD



Figure 4.4: Normals of the reconstructed mesh. Top: GOF, Bottom: Ours

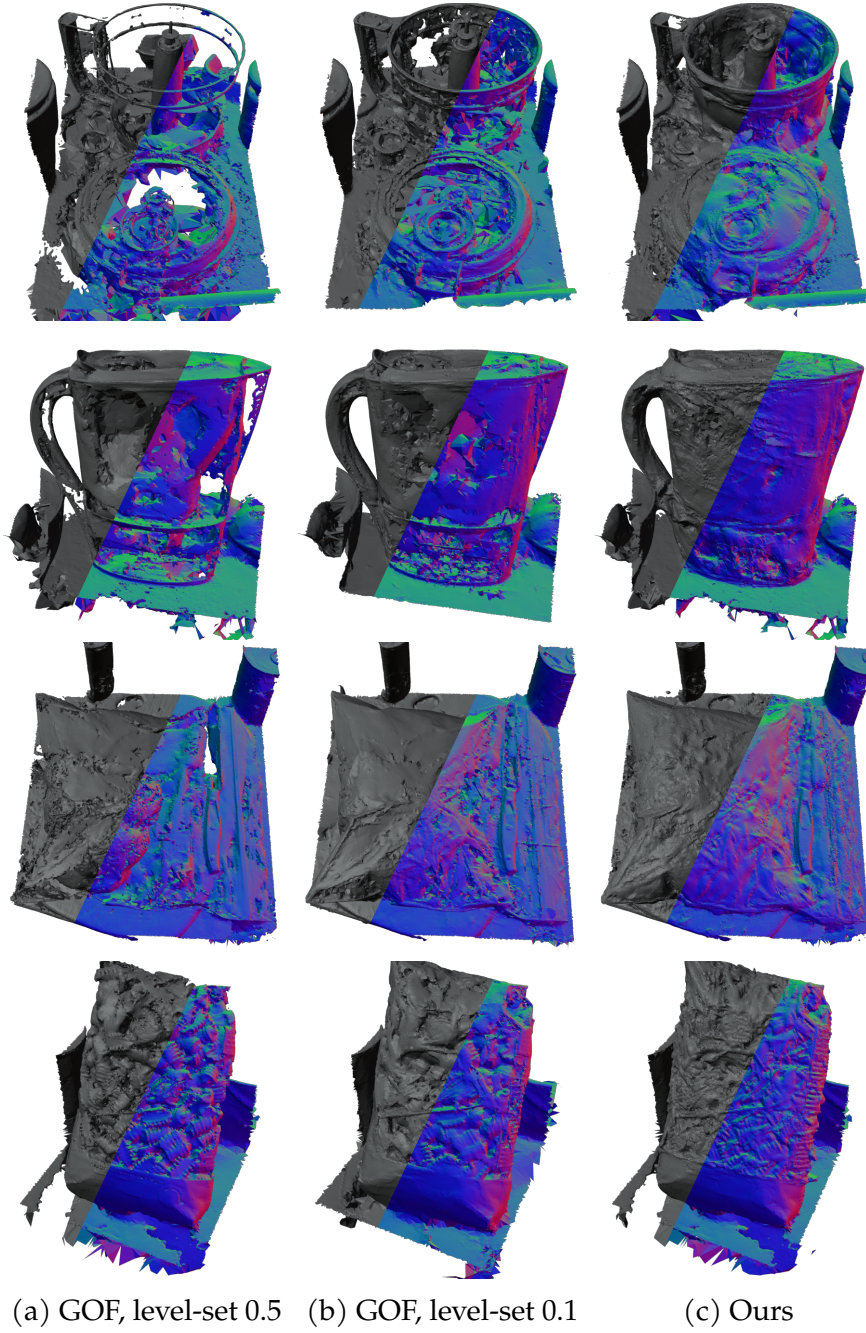


Figure 4.5: Visual comparison of transparent object surface reconstruction on the "Counter" scene of mip-NeRF 360 dataset.

4.5 Additional Experiment: SH-Based Segmentation

Prior to adopting a SAM-based segmentation approach, an alternative strategy was explored based on the spherical harmonic (SH) coefficients of the Gaussian primitives. The hypothesis was that the Y_{00} SH basis function encodes the view-independent (diffuse) component of color, while higher-order SH terms capture view-dependent effects, such as specular highlights. In this sense, the SH decomposition serves a role analogous to a Fourier transform on the sphere, with higher-order components corresponding to higher-frequency angular variation. It was therefore assumed that Gaussians corresponding to specular highlights common in transparent or glossy surfaces would exhibit greater energy in the higher-order SH coefficients. These coefficients could then be used to heuristically identify candidate Gaussians associated with transparent objects. Naive observations of the scene confirm that the SH coefficients of transparent objects do indeed capture specular reflections (see Fig.4.6).

We performed an experiment in which a scalar trait was calculated for each Gaussian primitive based on the SH coefficients corresponding to view-dependent effects (all coefficients excluding Y_{00}). The use of numerous metrics was attempted, including L1 and L2, as well as variants targeting only coefficients Y_l^m of $l = 1$ and $l = 2$ respectively.

While the method produces visually plausible results in some regions (see Fig.4.7), it produces both false positives and false negatives.

A possible explanation for this is the absence of regularization that encourages low-frequency encoding for diffuse surfaces. Furthermore, Gaussian primitives are occluded from most camera angles used in training. The view-independence of color for the turned away side of a partially occluded Gaussian primitive is not enforced. A diffuse Gaussian which is partially occluded may encode undesirable view-dependent effects on its turned-away side.

The SH encoding can also represent other view-dependent phenomena, such as ambient occlusion and shadowing apart from specular reflections. While a custom regularizer that biases diffuse regions toward low-order coefficients may improve our approach, it is unlikely to match the reliability and precision of SAM-based segmentation. Nonetheless, SH-based regularization remains an intriguing direction for future experiments.



Figure 4.6: Specular highlights of a transparent object captured in a real scene and encoded via SH coefficients. Viewing the Gaussians from a slightly tilted angle hides view-dependent specular reflections.

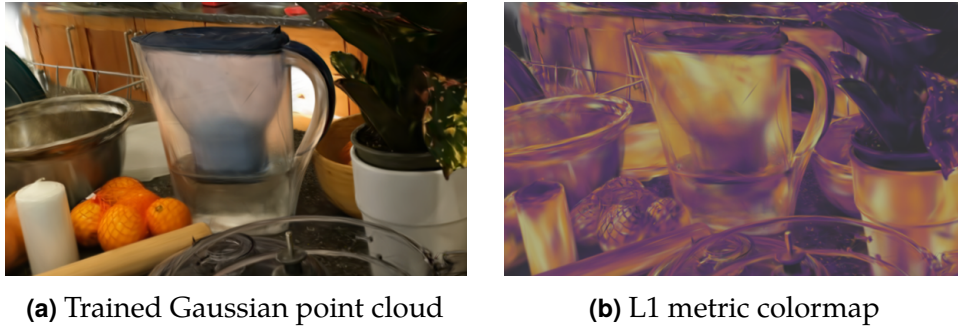


Figure 4.7: An example attempt of using SH coefficients for segmentation. The normalized L1 metric of SH coefficients Y_l^m such that $l > 0$ is color mapped to Gaussian primitives.

5 Conclusion and Future Work

This thesis has established the theoretical foundations necessary to understand modern novel view synthesis methods based on radiance fields. It further surveyed recent literature in surface reconstruction that builds upon these view synthesis approaches. Finally, it introduced novel strategies for reconstructing transparent surfaces by extending the generic surface reconstruction methods. Collectively, the theoretical component offers a comprehensive overview of the field and highlights emerging trends and techniques, and is a valuable resource for readers with a background in computer graphics who wish to understand and engage with radiance field-based surface reconstruction methods. It also offers an informative survey for those specifically interested in the reconstruction of transparent objects.

The practical contribution of this thesis is a proof-of-concept method that enhances the Gaussian Opacity Fields framework [63]. Our approach samples multiple training iterations of the pipeline and addresses the conflation between visual opacity and physical density in the Gaussian representation post-training. A qualitative comparison with the baseline demonstrates the improved performance of our method. The efficiency of this simple approach indicates the potential of sampling training iterations as an alternative to the use of geometric priors.

The thesis additionally discovers possible demand for a large-scale comparison of both implicit and explicit radiance field surface reconstruction methods across multiple datasets. The methods require non-trivial setup to run, both in terms of software dependencies and expensive hardware for training. Novel methods publish their results in limited comparisons against state of the art on a selection of datasets and only provide a handful of qualitative comparisons. The authors of the TNT dataset[27] provide a continually updated online leaderboard, in which novel methods compete in surface reconstruction. However, qualitative results are only provided in images of surface point clouds, not extracted meshes.

More importantly, there is the need for a new industry standard dataset. There are numerous datasets which commonly appear in evaluations of radiance field based methods:

5. CONCLUSION AND FUTURE WORK

- NeRF dataset [37]: The NeRF dataset contains iamges and ground truth geometry of simple synthetic scenes with transparent backgrounds. It was commonly used in evaluations of early implicit NVS methods, as well as evaluations of surface reconstruction. The dataset is too simple for evaluation of modern methods.
- DTU MVS [20]: This dataset contains images of real-world small-scale staged scenes captured in a laboratory setting, in addition to corresponding ground truth geometry captured by a laser scanner. However, for most modern methods, the dataset is too simple and largely solved.
- Mip-NeRF 360 dataset [2]: The dataset contains images of real-world complicated scenes, but does not provide ground truth geometry. It is commonly used for comparing methods of novel view synthesis. Methods of surface reconstruction often use it for qualitative evaluations.
- Tanks and Temples [27]: The TNT dataset was specifically designed for comparisons of surface reconstruction quality. It contains both images and ground truth geometry data of large real-world scenes. The geometry data was obtained using a professional laser scanner. The dataset is commonly used for the evaluation of NVS and surface reconstruction methods.

However, neither of the commonly used datasets provides ground truth geometry of transparent objects. Methods of surface reconstruction which try to solve the challenge of transparent surfaces must either rely on qualitative comparisons, or train on low quality, small synthetic datasets which are not widely acknowledged. A dataset containing real-world images of transparent objects accompanied by their ground truth geometry has the potential to become the industry standard in this evolving field.

I personally wish to pursue this line of work further. In cooperation with Skeletex Research s.r.o. [47], we aim to produce and publish such dataset, along with an evaluation of existing methods on the novel dataset.

A An appendix

The following files are being submitted with the thesis:

- The Gaussian point clouds used to produce meshes in our evaluation.
- The extracted meshes of GOF and our method.
- The Blender rendering setup and scripts used in preparation of figures.
- A large collection of images, comparing various training and adjustment configurations.

The GOF Docker image used for experiments on Aura is available at DockerHub: [silentbenefactor/gaussian-opacity-fields](https://hub.docker.com/r/silentbenefactor/gaussian-opacity-fields).

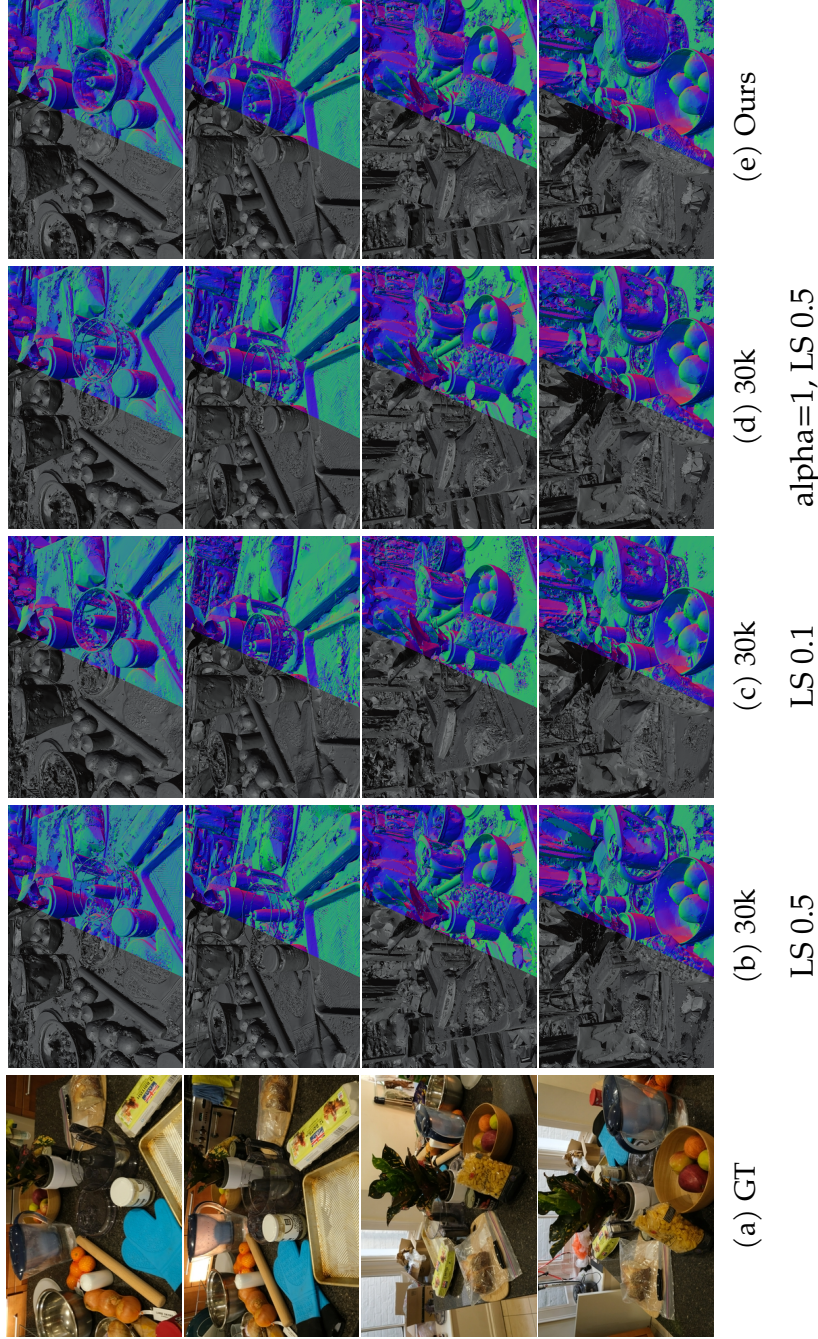


Figure A.1: Ablations of our method. Our method outperforms all of the attempted configurations of GOF. b), c) show difference choices of level sets, d) shows manual increase of Gaussian primitive opacity.

Bibliography

- [1] Aviral Agrawal et al. “Clear-Splatting: Learning Residual Gaussian Splats for Transparent Object Manipulation”. In: *RoboNerF: 1st Workshop On Neural Fields In Robotics at ICRA 2024*. 2024. URL: <https://openreview.net/forum?id=HcUC6hGcwu>.
- [2] Jonathan T. Barron et al. “Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 5460–5469. DOI: 10.1109/CVPR52688.2022.00539.
- [3] Jiazhong Cen et al. *Segment Any 3D Gaussians*. 2025. arXiv: 2312.00860 [cs.CV]. URL: <https://arxiv.org/abs/2312.00860>.
- [4] Anpei Chen et al. “TensorRF: Tensorial Radiance Fields”. In: *Computer Vision – ECCV 2022*. Ed. by Shai Avidan et al. Cham: Springer Nature Switzerland, 2022, pp. 333–350. ISBN: 978-3-031-19824-3.
- [5] Danpeng Chen et al. “PGSR: Planar-based Gaussian Splatting for Efficient and High-Fidelity Surface Reconstruction”. In: *IEEE Transactions on Visualization and Computer Graphics* (2024), pp. 1–12. ISSN: 2160-9306. DOI: 10.1109/tvcg.2024.3494046. URL: <http://dx.doi.org/10.1109/TVCG.2024.3494046>.
- [6] Runnan Chen et al. *OVGaussian: Generalizable 3D Gaussian Segmentation with Open Vocabularies*. 2024. arXiv: 2501.00326 [cs.CV]. URL: <https://arxiv.org/abs/2501.00326>.
- [7] Xiaoxue Chen et al. *NeRRF: 3D Reconstruction and View Synthesis for Transparent and Specular Objects with Neural Refractive-Reflective Fields*. 2023. arXiv: 2309.13039 [cs.CV]. URL: <https://arxiv.org/abs/2309.13039>.
- [8] Seokhun Choi et al. “Click-Gaussian: Interactive Segmentation to Any 3D Gaussians”. In: *Computer Vision – ECCV 2024*. Ed. by Aleš Leonardis et al. Cham: Springer Nature Switzerland, 2025, pp. 289–305. ISBN: 978-3-031-72646-0.
- [9] Brian Curless and Marc Levoy. “A Volumetric Method for Building Complex Models from Range Images”. In: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. ACM, 1996, pp. 303–312.

BIBLIOGRAPHY

- [10] Qiyu Dai et al. “GraspNeRF: Multiview-based 6-DoF Grasp Detection for Transparent and Specular Objects Using Generalizable NeRF”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 1757–1763. doi: 10.1109/ICRA48891.2023.10160842.
- [11] Bardienus P. Duisterhof et al. “Residual-NeRF: Learning Residual NeRFs for Transparent Object Manipulation”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. 2024, pp. 13918–13924. doi: 10.1109/ICRA57147.2024.10611224.
- [12] Sara Fridovich-Keil et al. “Plenoxels: Radiance Fields without Neural Networks”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 5491–5500. doi: 10.1109/CVPR52688.2022.00542.
- [13] Sarah F. Frisken et al. “Adaptively Sampled Distance Fields: A General Representation of Shape for Computer Graphics”. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 249–254.
- [14] Antoine Guédon. *3D Gaussian Splatting Tutorial (Part 4)*. International Conference on 3D Vision Tutorial. 2024. url: <https://3dgsutorial.github.io/>.
- [15] Antoine Guédon and Vincent Lepetit. “SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering”. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 5354–5363. doi: 10.1109/CVPR52733.2024.00512.
- [16] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.
- [17] Peter Hedman et al. “Baking Neural Radiance Fields for Real-Time View Synthesis”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 5855–5864. doi: 10.1109/ICCV48922.2021.00582.
- [18] Binbin Huang et al. “2D Gaussian Splatting for Geometrically Accurate Radiance Fields”. In: *ACM SIGGRAPH 2024 Conference Papers*. SIGGRAPH ’24. Denver, CO, USA: Association for Computing Machinery, 2024. isbn: 9798400705250. doi: 10.1145/3641519.3657428. url: <https://doi.org/10.1145/3641519.3657428>.

- [19] Jeffrey Ichnowski et al. “Dex-NeRF: Using a Neural Radiance Field to Grasp Transparent Objects”. In: *Proceedings of the 5th Conference on Robot Learning*. Ed. by Aleksandra Faust, David Hsu, and Gerhard Neumann. Vol. 164. Proceedings of Machine Learning Research. PMLR, Nov. 2022, pp. 526–536. URL: <https://proceedings.mlr.press/v164/ichnowski22a.html>.
- [20] Rasmus Jensen et al. “Large scale multi-view stereopsis evaluation”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2014, pp. 406–413.
- [21] Michael M. Kazhdan, Matthew Bolitho, and Hugues Hoppe. “Poisson Surface Reconstruction”. In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. Ed. by Alla Sheffer and Konrad Polthier. Vol. 256. SGP ’06. Cagliari, Sardinia, Italy: Eurographics Association, 2006, pp. 61–70. ISBN: 3-905673-36-3. URL: <http://dl.acm.org/citation.cfm?id=1281957.1281965>.
- [22] Bernhard Kerbl et al. “3D Gaussian Splatting for Real-Time Radiance Field Rendering”. In: *ACM Trans. Graph.* 42.4 (July 2023). ISSN: 0730-0301. DOI: 10.1145/3592433. URL: <https://doi.org/10.1145/3592433>.
- [23] Justin Kerr et al. “Evo-NeRF: Evolving NeRF for Sequential Robot Grasping of Transparent Objects”. In: *Proceedings of The 6th Conference on Robot Learning*. Ed. by Karen Liu, Dana Kulic, and Jeff Ichnowski. Vol. 205. Proceedings of Machine Learning Research. PMLR, Dec. 2023, pp. 353–367. URL: <https://proceedings.mlr.press/v205/kerr23a.html>.
- [24] Shakiba Kheradmand et al. “3D Gaussian Splatting as Markov Chain Monte Carlo”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., 2024, pp. 80965–80986. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/93be245fce00a9bb2333c17ceae4b732-Paper-Conference.pdf.
- [25] Jeongyun Kim et al. *TranSplat: Surface Embedding-guided 3D Gaussian Splatting for Transparent Object Manipulation*. 2025. arXiv: 2502.07840 [cs.CV]. URL: <https://arxiv.org/abs/2502.07840>.

BIBLIOGRAPHY

- [26] Alexander Kirillov et al. “Segment Anything”. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 3992–4003. doi: 10.1109/ICCV51070.2023.00371.
- [27] Arno Knapitsch et al. “Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction”. In: *ACM Transactions on Graphics* 36.4 (2017).
- [28] Peter Kravár. “Surface Reconstruction of Transparent Objects from Gaussian Radiance Fields”. In: *Proceedings of the Central European Seminar on Computer Graphics*. To appear. Smolenice, Slovakia, 2025.
- [29] Meenakshi Krishnan, Liam Fowl, and Ramani Duraiswami. *3D Gaussian Splatting with Normal Information for Mesh Extraction and Improved Rendering*. 2025. arXiv: 2501.08370 [cs.GR]. URL: <https://arxiv.org/abs/2501.08370>.
- [30] Congcong Li et al. *Car-GS: Addressing Reflective and Transparent Surface Challenges in 3D Car Reconstruction*. 2025. arXiv: 2501.11020 [cs.CV]. URL: <https://arxiv.org/abs/2501.11020>.
- [31] Mingwei Li et al. *TSGS: Improving Gaussian Splatting for Transparent Surface Reconstruction via Normal and De-lighting Priors*. 2025. arXiv: 2504.12799 [cs.CV]. URL: <https://arxiv.org/abs/2504.12799>.
- [32] Zhaoshuo Li et al. “Neuralangelo: High-Fidelity Neural Surface Reconstruction”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 8456–8465. doi: 10.1109/CVPR52729.2023.00817.
- [33] Siyun Liang et al. *SuperGSeg: Open-Vocabulary 3D Segmentation with Structured Super-Gaussians*. 2024. arXiv: 2412.10231 [cs.CV]. URL: <https://arxiv.org/abs/2412.10231>.
- [34] William E. Lorensen and Harvey E. Cline. “Marching Cubes: A High Resolution 3D Surface Construction Algorithm”. In: *ACM SIGGRAPH Computer Graphics* 21.4 (1987), pp. 163–169.
- [35] Xiaoyang Lyu et al. “3DGSR: Implicit Surface Reconstruction with 3D Gaussian Splatting”. In: *ACM Trans. Graph.* 43.6 (Nov. 2024). ISSN: 0730-0301. doi: 10.1145/3687952. URL: <https://doi.org/10.1145/3687952>.
- [36] Saswat Subhajyoti Mallick et al. “Taming 3DGS: High-Quality Radiance Fields with Limited Resources”. In: *SIGGRAPH Asia 2024 Conference Papers*. SA '24. Tokyo, Japan: Association for

- Computing Machinery, 2024. ISBN: 9798400711312. DOI: 10.1145/3680528.3687694. URL: <https://doi.org/10.1145/3680528.3687694>.
- [37] Ben Mildenhall et al. “NeRF: representing scenes as neural radiance fields for view synthesis”. In: *Commun. ACM* 65.1 (Dec. 2021), pp. 99–106. ISSN: 0001-0782. DOI: 10.1145/3503250. URL: <https://doi.org/10.1145/3503250>.
- [38] Thomas Müller et al. “Instant Neural Graphics Primitives with a Multiresolution Hash Encoding”. In: *ACM Trans. Graph.* 41.4 (July 2022), 102:1–102:15. DOI: 10.1145/3528223.3530127. URL: <https://doi.org/10.1145/3528223.3530127>.
- [39] Richard A. Newcombe et al. “KinectFusion: Real-time dense surface mapping and tracking”. In: *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. 2011, pp. 127–136. DOI: 10.1109/ISMAR.2011.6092378.
- [40] Michael Oechsle, Songyou Peng, and Andreas Geiger. “UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 5569–5579. DOI: 10.1109/ICCV48922.2021.00554.
- [41] Lukas Radl et al. “StopThePop: Sorted Gaussian Splatting for View-Consistent Real-time Rendering”. In: *ACM Trans. Graph.* 43.4 (July 2024). ISSN: 0730-0301. DOI: 10.1145/3658187. URL: <https://doi.org/10.1145/3658187>.
- [42] Nasim Rahaman et al. “On the Spectral Bias of Neural Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 5301–5310. URL: <https://proceedings.mlr.press/v97/rahaman19a.html>.
- [43] Marie-Julie Rakotosaona et al. “NeRFMeshing: Distilling Neural Radiance Fields into Geometrically-Accurate 3D Meshes”. In: *International Conference on 3D Vision, 3DV 2024, Davos, Switzerland, March 18-21, 2024*. IEEE, 2024, pp. 1156–1165. ISBN: 979-8-3503-6245-9. DOI: 10.1109/3DV62453.2024.00093. URL: <https://doi.org/10.1109/3DV62453.2024.00093>.
- [44] Christian Reiser et al. “KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs”. In: *2021 IEEE/CVF Inter-*

BIBLIOGRAPHY

- national Conference on Computer Vision (ICCV)*. 2021, pp. 14315–14325. doi: 10.1109/ICCV48922.2021.01407.
- [45] Johannes L. Schönberger and Jan-Michael Frahm. “Structure-from-Motion Revisited”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4104–4113. doi: 10.1109/CVPR.2016.445.
- [46] Johannes L. Schönberger et al. “Pixelwise View Selection for Unstructured Multi-View Stereo”. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 501–518. ISBN: 978-3-319-46487-9.
- [47] *Skeletex Research Group*. 2025. URL: <https://skeletex.xyz/home> (visited on 05/20/2025).
- [48] Cheng Sun, Min Sun, and Hwann-Tzong Chen. “Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 5449–5459. doi: 10.1109/CVPR52688.2022.00538.
- [49] Jia-Mu Sun et al. “NU-NeRF: Neural Reconstruction of Nested Transparent Objects with Uncontrolled Capture Environment”. In: *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2024)* 43.6 (2024). doi: 10.1145/3687757.
- [50] Graham M. Treece, Richard W. Prager, and Andrew H. Gee. “Volume Rendering of Multiple Intersecting Surfaces using Marching Tetrahedra”. In: *Medical Image Analysis* 4.3 (1999), pp. 315–324.
- [51] Matias Turkulainen et al. “DN-Splatter: Depth and Normal Priors for Gaussian Splatting and Meshing”. In: *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*. Feb. 2025, pp. 2421–2431.
- [52] Dor Verbin et al. “Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields”. In: *CVPR* (2022).
- [53] Jiepeng Wang et al. “GausSurf: Geometry-Guided 3D Gaussian Splatting for Surface Reconstruction”. In: *arXiv preprint arXiv:2411.19454* (2024).
- [54] Peng Wang et al. “NeuS: learning neural implicit surfaces by volume rendering for multi-view reconstruction”. In: *Proceedings of the 35th International Conference on Neural Information Processing*

- Systems*. NIPS '21. Red Hook, NY, USA: Curran Associates Inc., 2021. ISBN: 9781713845393.
- [55] Lee A Westover. *SPLATTING: A Parallel, Feed-Forward Volume Rendering Algorithm*. Tech. rep. USA, 1991.
 - [56] Qianyi Wu, Jianmin Zheng, and Jianfei Cai. “Surface Reconstruction from 3D Gaussian Splatting via Local Structural Hints”. In: *Computer Vision – ECCV 2024*. Ed. by Aleš Leonardis et al. Cham: Springer Nature Switzerland, 2025, pp. 441–458. ISBN: 978-3-031-72627-9.
 - [57] Tianhao Walter Wu et al. “AlphaSurf: Implicit Surface Reconstruction for Semi-Transparent and Thin Objects with Decoupled Geometry and Opacity”. In: *3DV*. 2025.
 - [58] Yanmin Wu et al. *OpenGaussian: Towards Point-Level 3D Gaussian-based Open Vocabulary Understanding*. 2024. arXiv: 2406.02058 [cs.CV]. URL: <https://arxiv.org/abs/2406.02058>.
 - [59] Qiangeng Xu et al. “Point-NeRF: Point-based Neural Radiance Fields”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 5428–5438. DOI: 10.1109/CVPR52688.2022.00536.
 - [60] Ziyi Yang et al. “Spec-Gaussian: Anisotropic View-Dependent Appearance for 3D Gaussian Splatting”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., 2024, pp. 61192–61216. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/708e0d691a22212e1e373dc8779cbe53-Paper-Conference.pdf.
 - [61] Lior Yariv et al. “BakedSDF: Meshing Neural SDFs for Real-Time View Synthesis”. In: *ACM SIGGRAPH 2023 Conference Proceedings*. SIGGRAPH '23. Los Angeles, CA, USA: Association for Computing Machinery, 2023. ISBN: 9798400701597. DOI: 10.1145/3588432.3591536. URL: <https://doi.org/10.1145/3588432.3591536>.
 - [62] Alex Yu et al. “PlenOctrees for Real-Time Rendering of Neural Radiance Fields”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 5732–5741. DOI: 10.1109/ICCV48922.2021.00570.

BIBLIOGRAPHY

- [63] Zehao Yu, Torsten Sattler, and Andreas Geiger. “Gaussian Opacity Fields: Efficient Adaptive Surface Reconstruction in Unbounded Scenes”. In: *ACM Trans. Graph.* 43.6 (Nov. 2024). ISSN: 0730-0301. DOI: 10 . 1145 / 3687937. URL: [https : //doi .org/10.1145/3687937](https://doi.org/10.1145/3687937).
- [64] Ziyu Zhang et al. *Quadratic Gaussian Splatting for Efficient and Detailed Surface Reconstruction*. 2024. arXiv: 2411.16392 [cs.CV]. URL: <https://arxiv.org/abs/2411.16392>.
- [65] Matthias Zwicker et al. “Surface splatting”. In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’01. New York, NY, USA: Association for Computing Machinery, 2001, pp. 371–378. ISBN: 158113374X. DOI: 10.1145/383259.383300. URL: <https://doi.org/10.1145/383259.383300>.