

Point-Based 3D Reconstruction from Sparse Views under Known Illumination

Magnus Kaufmann Gjerde¹, Joakim Bruslund Haurum^{2,5}, Jeppe Revall Frisvad³, Markus Worchel⁴, J. Andreas Bærentzen³, and Thomas B. Moeslund^{1,5}

¹ Visual Analysis & Perception Lab, Aalborg University

² Centre for Software Technology, University of Southern Denmark

³ Department of Applied Mathematics and Computer Science, Technical University of Denmark

⁴ Technische Universität Berlin

⁵ Pioneer Centre for Artificial Intelligence

Abstract. Sparse view 3D reconstruction is commonly addressed with neural implicit surfaces or dense point-based representations such as Gaussian splatting. Surface-aware splatting methods improve extracted geometry through oriented primitives and regularization, while RadiosityGS incorporates differentiable light transport through a radiosity inspired finite-element surfel formulation. We propose a differentiable point rendering method based on opacity-bearing beta surfels. An opacity explicit adjoint light transport formulation provides gradients for surfel geometry and appearance parameters, allowing physically based light transport to constrain reconstruction. Across five synthetic objects reconstructed from ten posed views, our method achieves the lowest mean symmetric Chamfer distance among the evaluated baselines and reduces mean Chamfer distance by 28.5% relative to the strongest point-based baseline while using only 267 surfels on average, approximately $161\times$ fewer primitives. Directional Chamfer results further show improved accuracy and competitive completion relative to related point-based methods. These results show that, in the controlled direct illumination setting, compact beta surfels combined with transport-based optimization can recover surfaces without relying on the tens to hundreds of thousands of primitives used by the evaluated baselines.

Keywords: Differentiable rendering · Point-based reconstruction · Inverse rendering · Adjoint light transport · Gaussian splatting

1 Introduction

Recovering accurate geometry from a small set of images remains challenging because appearance alone does not uniquely determine surface shape, reflectance, and visibility. Recent point-based methods, such as Gaussian Splatting (GS) [7], can synthesize high-quality images by optimizing large sets of primitives. However, image fidelity does not necessarily imply accurate geometry: Optimized

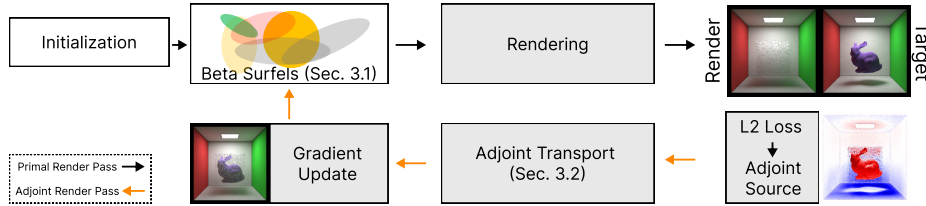


Fig. 1: Overview of our optimization pipeline. Starting from a sparse beta-surfel initialization, we alternate primal rendering, residual evaluation, adjoint transport, and parameter updates. The adjoint source is the image-space residual between the rendered and reference images.

primitives may compensate for photometric error through opacity and their appearance parameters rather than converging to a coherent surface. Consequently, meshes extracted from appearance-driven splatting representations can contain floaters, fragmented regions, and excessive geometric detail. Surface-aware variants such as 2DGS [4] and SuGaR [3] improve geometric behavior through oriented primitives and regularization, but they still rely on rasterization-based image formation rather than physically based light transport.

Known illumination provides an additional source of geometric supervision. Under calibrated point lighting and Lambertian reflectance, surface shape and orientation affect observed radiance through shading and visibility. Physics-based differentiable rendering provides a principled way to exploit these effects by differentiating image formation through physical light transport [10, 24].

RadiosityGS [5] combines Gaussian surfels with a radiosity-inspired finite-element formulation of differentiable light transport for relighting and geometry reconstruction. Its formulation represents radiometric quantities using per-surfel coefficients and assumes spatially constant material, emission, and outgoing radiance within each Gaussian-surfel support. This enables efficient view-independent transport computation and motivates our alternative formulation, which, in contrast, expresses transport through surface-level quantities evaluated at the specific local coordinates of each surfel interaction, allowing radiance and material response to vary across the surfel support.

We adopt this local-coordinate view for compact surface recovery. Our representation uses opacity-bearing beta surfels [11] with spatially varying local support. Rather than assigning one outgoing radiance value to an entire primitive, we evaluate scattering, transmittance, and visibility at sampled surfel coordinates. We then specialize continuous adjoint light transport to this representation, retaining segment transmittance explicitly to obtain analytical parameter gradients for surfel geometry, material, opacity, and local opacity-profile shape. We estimate the resulting transport integrals with Monte Carlo path sampling and evaluate the direct illumination specialization under calibrated point lights.

Figure 1 summarizes the optimization loop. Starting from a sparse beta-surfel initialization, primal rendering produces image residuals that define the adjoint

source. The adjoint transport pass then provides gradients used to update surfel geometry, material, opacity, and kernel shape.

We evaluate the method on five synthetic objects reconstructed from ten posed input views under calibrated static near-field point illumination. Our method obtains the lowest mean symmetric Chamfer distance among the evaluated baselines while using only 267 surfels on average. In comparison, the standard 2DGS 30K configuration uses 51k primitives on average. Directional Chamfer measurements show the lowest reconstruction-to-ground-truth error and competitive ground-truth-to-reconstruction error. Although we focus on isolated objects rather than large-scale scenes, we address object-centric 3D reconstruction by recovering geometry with orders of magnitude fewer primitives than the evaluated point-based baselines while maintaining better reconstruction results.

Our contributions are:

- A method for sparse-view reconstruction under known direct illumination using compact, opacity-bearing beta surfels, with explicit scattering, visibility, and transmittance and without assuming constant outgoing radiance over each surfel.
- An opacity-explicit adjoint light-transport formulation that provides gradients with respect to surfel geometry, albedo, opacity, and local opacity-profile shape.

2 Related Work

Physics-based differentiable rendering. This approach optimizes scene parameters by differentiating image formation with respect to geometry, materials, or illumination. Early Monte Carlo methods developed estimators for visibility changes and transport derivatives under complex light transport [10, 23, 24]. Adjoint formulations provide an alternative perspective in which gradients are derived at the level of the continuous transport operator before discretization [13, 15]. Recent work has extended differentiable light transport to point-based scene representations. In particular, RadiosityGS [5] formulates transport over Gaussian surfels using a radiosity-inspired finite-element approximation. In contrast, our method instead evaluates transport using generic Monte Carlo path tracing, without a constant local radiance assumption.

Surface reconstruction from posed images. Neural implicit methods such as NeuS [18] and NeuS2 [19] optimize continuous implicit surfaces from posed images, while GeoSVR [9] uses sparse voxels and geometry-oriented regularization. These methods contextualize our explicit surfel representation, which directly exploits calibrated point-light transport.

Point-based reconstruction and Gaussian splatting. Point-based rendering has a long history in graphics, and Gaussian splatting methods have recently enabled high-quality novel-view synthesis with optimized primitive sets [7, 25]. Surface-aware variants improve geometric behavior by using oriented primitives

or additional regularization. In particular, 2DGS [4] represents scenes using oriented 2D Gaussian disks and introduces depth-distortion and normal-consistency losses for mesh reconstruction. SuGaR [3] aligns Gaussian primitives with an extracted surface to improve mesh quality. Our representation is similarly surface-oriented, but differs in its compact-support opacity kernel, continuous transmittance model, and physics-based optimization under known illumination.

3 Method

Our goal is to recover scene parameters ψ (geometry, materials, etc.), whose rendered measurements match the posed reference images under known direct illumination, while evaluating transport at local surfel coordinates rather than assuming constant radiance over each support. We first discuss how we define our geometry primitive before proceeding to differentiate light transport with respect to the scene parameters.

3.1 Point Geometry Parameterization

We introduce our beta surfel elements by expressing them as regular surfaces. We do so by using the theory of Do Carmo [2]. A beta surfel represents a finite planar surface element embedded in $\mathbb{R}^3$ and parameterized by a local tangent frame. Each surfel k is defined by a learnable set of parameters:

$$\psi_k = \{\mathbf{p}_k, \mathbf{t}_{u,k}, \mathbf{t}_{v,k}, \mathbf{s}_k, \rho_k, \eta_k, \beta_k\}, \quad (1)$$

where $\mathbf{p}_k \in \mathbb{R}^3$ is the surfel center, $\mathbf{t}_{u,k}$ and $\mathbf{t}_{v,k}$ are orthonormal tangent directions, defining the surfel normal by $\mathbf{n}_k = \mathbf{t}_{u,k} \times \mathbf{t}_{v,k}$, $\mathbf{s}_k = (s_{u,k}, s_{v,k})$ are scale parameters for the tangent directions, ρ_k is a diffuse albedo, $\eta_k \in [0, 1]$ is a uniform opacity factor, and β_k controls the shape of the opacity kernel.

We define a local chart of the regular surface:

$$\Phi_k(u, v) = \mathbf{p}_k + s_{u,k} \mathbf{t}_{u,k} u + s_{v,k} \mathbf{t}_{v,k} v, \quad (2)$$

with $(u, v) \in \mathbb{R}^2$ as the local coordinates of the surfel. We also define a constant Jacobian determinant which is found by analyzing the surface with its first fundamental form

$$dA = s_{u,k} s_{v,k} du dv, \quad (3)$$

which we use for reparameterizing the transport operator integral over beta surfels.

We use the local coordinates (u, v) to model continuous visibility and transmittance α_k^{geom} . While Gaussian kernels are commonly used in point-based rendering, we instead use a beta kernel presented by Liu *et al.* [11]:

$$\alpha_k^{\text{geom}}(u, v; \beta_k) = \begin{cases} (1 - r^2)^{b(\beta_k)}, & r^2 \leq 1, \\ 0, & r^2 > 1, \end{cases} \quad (4)$$

with

$$b(\beta_k) = 4e^{\beta_k} \quad \text{and} \quad r^2 = u^2 + v^2. \quad (5)$$

When $\beta_k = 0$ the kernel starts with a Gaussian-like function. The final effective opacity of the surfel is

$$\alpha_k(u, v) = \eta_k \alpha_k^{\text{geom}}(u, v; \beta_k), \quad (6)$$

where $0 \leq \alpha_k \leq 1$. Later in ablation studies it is shown that optimizing the beta profile improves reconstruction on our data relative to the fixed Gaussian-like $\beta_k = 0$ profile. Furthermore, a benefit of the compactly supported beta kernel is that it removes the need for a heuristic footprint cutoff which can vary between different Gaussian Splatting implementations.

As shown with our definition of opacity, a beta surfel is not an opaque surface element. Its local opacity profile both determines scattering at a sampled point and attenuates every light or camera connection passing through its support. Consequently, changing a surfel parameter affects image formation in two ways, either through local scattering or segment transmittance. Monte Carlo implementations can account for opacity through stochastic path continuation which is sufficient for rendering, but it does not directly expose the derivatives of the continuous transmittance needed to differentiate analytically. We therefore retain a segment transmittance defined as τ_ψ explicitly before forming the light transport gradients.

This also differs from the primitive-level finite-element discretization used by RadiosityGS [5], which represents radiometric quantities using per-surfel coefficients and assumes them spatially constant within each Gaussian-surfel support. In contrast, we estimate transport at sampled local coordinates within each beta-surfel support.

3.2 Light Transport Gradients with Explicit Transmittance

The gradients considered here could alternatively be obtained by applying reverse-mode automatic differentiation to a sampled rendering computation. While this can be practical at first, differentiating recursive transport becomes increasingly demanding as longer path histories and their intermediate states must be stored. We instead follow the adjoint perspective of Radiative Backpropagation [13, 15], while retaining explicit transmittance τ_ψ . We subsequently show by example the zeroth-order term of the adjoint Neumann series to show that an appropriate parameterization of the transport operator keeps the sensor sample fixed, thereby avoiding camera-response derivatives induced by a moving surface projection.

We optimize scene parameters ψ for all surfels described in the previous section, to obtain a rendered image matching an observed reference image. We write beam transmittance as τ_ψ in the rendering equation, where the ψ subscript denotes the dependence on the scene parameters ψ . This makes the transmittance terms involving $\partial\tau_\psi/\partial\psi$ explicit and connects the continuous adjoint formulation to our opacity-bearing point primitives. Assuming a pinhole camera at the position $\mathbf{x}_c$, the intensity captured in one of its pixels is an integral of incident

radiance L_i across points $\mathbf{x}_p$ in the projected pixel area, or equivalently an integral of outgoing radiance L_o from observed surface positions $\mathbf{x}_o$ found in the ray direction $\boldsymbol{\omega} = (\mathbf{x}_p - \mathbf{x}_c)/t_p$ with $t_p = \|\mathbf{x}_p - \mathbf{x}_c\|$, such that $\mathbf{x}_p = \mathbf{x}_c + t_p \boldsymbol{\omega}$ and $\mathbf{x}_o = \mathbf{x}_c + t_o \boldsymbol{\omega}$. A least squares cost function for our optimization is then

$$\mathcal{J}(L_o, \psi) = \int_{A(\psi)} \int_{4\pi} W_c(\mathbf{x}, \boldsymbol{\omega}) J(L_o, \psi) d\boldsymbol{\omega} dA \quad (7)$$

$$J(L_o, \psi) = \frac{1}{2} (L_o(\mathbf{x}_p, -\boldsymbol{\omega}, \psi) - L_{\text{ref}}(\mathbf{x}_p, -\boldsymbol{\omega}))^2. \quad (8)$$

where $\mathbf{x}$ is a point in $A(\psi)$, which is the union of $\mathbf{x}_c$ and all surface positions in the scene. The domain 4π refers to the full solid angle of directions $\boldsymbol{\omega}$, L_{ref} is radiance sampled in $\mathbf{x}_p$ of the reference image, and $W_c = [\boldsymbol{\omega} \in \Omega] \delta(\mathbf{x} - \mathbf{x}_c)$ is the camera response function, defined using an Iverson bracket and a Dirac delta function to only include paths observed at $\mathbf{x}_c$ with $\boldsymbol{\omega}$ in the solid angle Ω subtended by the camera sensor (the film). The optimal parameters are obtained by solving

$$\psi^* = \arg \min_{\psi} \mathcal{J}(L_o, \psi) \quad \text{s.t.} \quad R(L_o, \psi) = 0, \quad (9)$$

where the constraint function R is based on light transport.

In the original discussion of methods for solving the rendering equation, Kajiya [6] mentioned that one option is to use a linear operator $\mathcal{T}_\psi$ to represent the integral in the reflected radiance term. We write the rendering equation including transmittance as:

$$L_o = \tau_\psi (L_e + \mathcal{T}_\psi L_o), \quad (10)$$

where L_e is emitted radiance and τ_ψ is beam transmittance induced by the opacity-bearing scene representation. This is our main addition to Stam's formulation [15]. The transport operator $\mathcal{T}_\psi$ acts on outgoing radiance and includes propagation and scattering between surface elements. Writing τ_ψ explicitly is not required in all presentations of light transport, but it is necessary here because the opacity parameters of the point primitives are optimized and therefore contribute directly to the gradient.

Using Lagrange multipliers and the continuous adjoint representation we end up with the following expression (more details in supplemental Appendix A):

$$\frac{\partial \mathcal{L}}{\partial \psi} = \frac{\partial \mathcal{J}}{\partial \psi} + \left\langle \frac{\partial(\mathcal{T}_\psi^* \tau_\psi)}{\partial \psi} p, L_o \right\rangle + \left\langle p, \frac{\partial \tau_\psi}{\partial \psi} L_e \right\rangle, \quad (11)$$

where $*$ denotes the adjoint and p is a Lagrangian multiplier traced as incident directional importance. The appeal of the adjoint formulation is that it eliminates the need to compute the variation of L_o with the parameters ψ explicitly. We only need derivatives of the transport operator kernel for each surfel, yielding favorable scaling properties with respect to the number of optimized parameters.

For a fixed camera and camera response function W_c , the measurement operator and the image domain are independent of the scene parameters ψ . Consequently, the cost functional $\mathcal{J}$ has no explicit dependence on ψ , and the first term of Eq. (11) vanishes.

3.3 Transport and Transmittance Derivatives

As derived in Sec. A.1 of the supplementary material, the Lagrangian multiplier p is given as a Neumann series:

$$p = \sum_{n=0}^{\infty} (\mathcal{T}_{\psi}^* \tau_{\psi})^n \frac{\partial J}{\partial L_o} W_c = \sum_{n=0}^{\infty} (\mathcal{T}_{\psi}^* \tau_{\psi})^n (L_o - L_{\text{ref}}) W_c.$$

To clearly show how the gradient connects to propagation of this adjoint variable, we take a special look at the zeroth-order term of this Neumann series, and denote it by

$$p_0 = (\mathcal{T}_{\psi}^* \tau_{\psi})^0 (L_o - L_{\text{ref}}) W_c = (L_o - L_{\text{ref}}) W_c. \quad (12)$$

Using the zeroth-order adjoint approximation $p_0 \approx p$, the gradient of the Lagrangian $\mathcal{L}$ with respect to the scene parameters ψ is then

$$\begin{aligned} \left. \frac{\partial \mathcal{L}}{\partial \psi} \right|_{p \approx p_0} &= \left\langle \frac{\partial (\mathcal{T}_{\psi}^* \tau_{\psi})}{\partial \psi} p_0, L_o \right\rangle + \left\langle p_0, \frac{\partial \tau_{\psi}}{\partial \psi} L_e \right\rangle. \\ &= \left\langle \frac{\partial \mathcal{T}_{\psi}^*}{\partial \psi} \tau_{\psi} p_0, L_o \right\rangle + \left\langle p_0, \frac{\partial \tau_{\psi}}{\partial \psi} (L_e + \mathcal{T}_{\psi} L_o) \right\rangle. \end{aligned} \quad (13)$$

The first term includes derivatives of transport from the camera to the scene. The second term includes derivatives of the transmittance of directly visible emitters and surfaces. In the following, we describe the computation of the transport and transmittance derivatives.

Transmittance derivative. Here, $\alpha_i(u_i, v_i)$ is the opacity of a surfel, and i is the i -th surfel in the front-to-back composition between $\mathbf{x}_0$ and $\mathbf{x}_1$:

$$\tau_{\psi}(\mathbf{x}_0, \mathbf{x}_1) = \prod_{i \in \mathcal{H}} (1 - \alpha_i(u_i, v_i)). \quad (14)$$

The transmittance derivative, where $\mathcal{H}$ is the fixed set of segment intersections assuming the ordering and membership of intersected surfels remain unchanged on the path between $\mathbf{x}_0$ and $\mathbf{x}_1$, is

$$\frac{\partial \tau_{\psi}(\mathbf{x}_0, \mathbf{x}_1)}{\partial \psi} = - \sum_{k \in \mathcal{H}} \frac{\partial \alpha_k(u_k, v_k)}{\partial \psi} \prod_{\substack{i \in \mathcal{H} \\ i \neq k}} (1 - \alpha_i(u_i, v_i)). \quad (15)$$

Transport derivative. The adjoint transport operator admits equivalent hemispherical and surface-area parameterizations. Here, $\boldsymbol{\theta}$ denotes the direction at which the operator is evaluated. In the hemispherical form, ϕ parameterizes the

direction from $\mathbf{x}_0$ and determines the next interaction $\mathbf{x}_1 = r_\psi(\mathbf{x}_0, \phi)$. The two forms are

$$\mathcal{T}_\psi^* p_0 = \int_{\mathbb{S}^2} p_0(r_\psi(\mathbf{x}_0, \phi), -\phi) f_s(\mathbf{x}_0, \boldsymbol{\theta}, -\phi) |\cos(\mathbf{N}_{\mathbf{x}_0}, \phi)| d\omega_\phi, \quad (16)$$

$$\mathcal{T}_\psi^* p_0 = \int_A p_0(\mathbf{x}_1, -\phi) f_s(\mathbf{x}_0, \boldsymbol{\theta}, -\phi) G(\mathbf{x}_0, \mathbf{x}_1) dA_{\mathbf{x}_1}. \quad (17)$$

For primal rendering, either formulation may be used since both represent the same transport operator. Under differentiation, however, the choice of parameterization becomes important, as discussed by Worchel *et al.* [20]. We use different parameterizations for different path segments: Recursive surface-to-surface transport is evaluated using the surface-area form, which permits reparameterization over fixed local surfel coordinates, while the primary camera-to-scene connection is evaluated using the hemispherical form. For recursive surface-to-surface transport, we locally denote the current interaction by $\mathbf{x}_0$ and the next interaction by $\mathbf{x}_1 = \Phi_k(u, v)$. Its derivative is

$$\frac{\partial \mathcal{T}_\psi^*}{\partial \psi} p_0 = \int_U \partial_\psi [p_0(\mathbf{x}_1, -\phi) f_s(\mathbf{x}_0, \boldsymbol{\theta}, -\phi) G(\mathbf{x}_0, \mathbf{x}_1) J_\psi] du dv. \quad (18)$$

Here, U denotes the local surfel-coordinate domain and $J_\psi = s_u s_v$ is the corresponding determinant of the surface-area Jacobian. For the primary camera-to-scene propagation, we instead differentiate the hemispherical form. Let $\mathbf{x}_1 = r_\psi(\mathbf{x}_c, \phi)$, where the camera position $\mathbf{x}_c$ and sampled sensor direction $\boldsymbol{\theta}$ are held fixed:

$$\frac{\partial \mathcal{T}_\psi^*}{\partial \psi} p_0 = \int_{\mathbb{S}^2} \partial_\psi [p_0(\mathbf{x}_1, -\phi) f_s(\mathbf{x}_1, \boldsymbol{\theta}, -\phi) |\cos(\mathbf{N}_{\mathbf{x}_1}, \phi)|] d\omega_\phi. \quad (19)$$

A surface-area parameterization would instead evaluate the adjoint source at the projection of a parameter-dependent surface point, $p_0(\mathbf{X}_\psi(u, v), \boldsymbol{\omega}_\psi(u, v))$. Differentiating this expression introduces chain-rule terms through the parameter-dependent evaluation of p_0 , for example $\nabla_{\mathbf{x}} p_0 \frac{\partial \mathbf{X}_\psi}{\partial \psi} + \nabla_{\boldsymbol{\omega}} p_0 \frac{\partial \boldsymbol{\omega}_\psi}{\partial \psi}$, analogous to the pixel-filter derivative discussed by Yu *et al.* [22]. For a discontinuous box reconstruction filter, this also introduces the associated pixel-boundary contribution. By keeping the sampled camera ray fixed, the hemispherical parameterization evaluates the reconstruction filter at a fixed sensor sample. The geometry dependence is instead represented by the moving camera-ray intersection $\mathbf{x}_1 = r_\psi(\mathbf{x}_c, \phi)$.

4 Problem Setup and Training

The preceding formulation is written for recursive light transport and therefore admits multi-bounce path contributions. In this work, however, we instantiate its direct-illumination specialization for sparse-view geometry reconstruction. Given

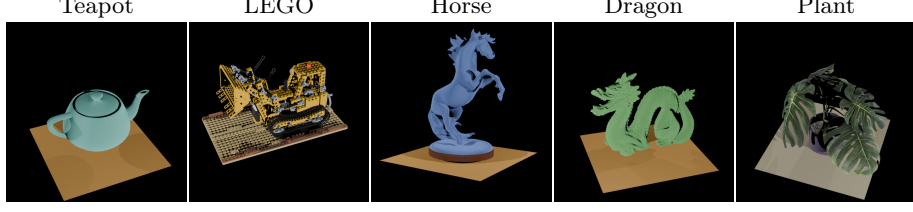


Fig. 2: Overview of the synthetic sparse-view object reconstruction dataset. Each scene is reconstructed from ten posed input views rendered with known static point-light illumination and Lambertian materials. For visual clarity, we show one representative training image per scene.

posed reference images, calibrated camera intrinsics and extrinsics, and known static point lights, we optimize the beta-surfel parameters ψ_k defined in Eq. (1).

The optimized objective combines image reconstruction with geometric regularization:

$$\mathcal{L}_{\text{train}} = \mathcal{J}_{\text{rgb}} + \lambda_d \mathcal{L}_d + \lambda_n \mathcal{L}_n + \lambda_{\text{weak}} \mathcal{L}_\alpha. \quad (20)$$

where $\mathcal{J}_{\text{rgb}}$ is the image reconstruction loss across all RGB channels, $\mathcal{L}_d$ is the depth-distortion regularizer, $\mathcal{L}_n$ is the normal-consistency regularizer adopted from 2DGS [4], and $\mathcal{L}_\alpha$ is a weak visibility-weighted opacity regularizer.

Weak Opacity Regularizer A black background introduces an opacity/radiance ambiguity where a partially transparent surfel with high reflected radiance can produce a similar image contribution to a more opaque surfel with lower reflected radiance. We therefore introduce a weak visibility-weighted regularizer that encourages opacity only for surfels that contribute to observed training pixels:

$$\mathcal{L}_\alpha = \sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{H}(p)} w_{p,i} (1 - \eta_i)^2, \quad (21)$$

where $\mathcal{P}$ denotes the set of training pixel rays and $\mathcal{H}(p)$ is the front-to-back ordered list of surfels intersected by ray p . The visibility weight is

$$w_{p,i} = \tau_{p,i} \alpha_{p,i}, \quad (22)$$

where $\tau_{p,i}$ is the accumulated transmittance before surfel i and $\alpha_{p,i}$ is its effective opacity at the ray-surfel intersection. We stop gradients through $w_{p,i}$ when evaluating this prior. The regularizer therefore cannot be reduced by changing the ray composition and instead directly encourages the opacity parameter of contributing surfels. The weight λ_{weak} in Eq. (20) is kept small, so the prior is intended to regularize the opacity/radiance ambiguity while retaining a small relative weight.

Densification and Pruning We adapt the number of surfels during optimization using clone-only densification based on position gradient and scale-based pruning. A surfel that is incorrectly positioned can reduce its image-space and light-transport contribution by shrinking its footprint. We therefore remove surfels whose scale falls below a fixed minimum threshold, as these primitives have become effectively unsupported by the reconstruction objective.

We do not use opacity pruning or periodic opacity resets commonly used by GS methods [7]. Given a background color in the renderer, a surfel contribution is jointly controlled by opacity and reflected radiance, so reducing opacity can mimic changes in albedo or incident illumination. In preliminary experiments, opacity resets and opacity-based pruning removed useful surfels or encouraged low-opacity configurations that destabilized the normal consistency regularizer. We therefore avoid resetting opacity and remove unsupported surfels only through scale-based pruning.

4.1 Dataset, Baselines, and Evaluation

We evaluate on five synthetic object scenes: *Teapot* [17], *LEGO* [12], *Dragon* [1], *Horse*, and *Plant*. The *Horse* and *Plant* assets were obtained from Blendkit under their respective licenses. We release only rendered data, calibration, and scene-generation code, not the original meshes or textures. Figure 2 shows one representative training image per scene. Each object is reconstructed from ten posed images at 500×500 resolution. The camera intrinsics and extrinsics, together with the positions and intensities of three to four static near-field point lights, are known to our method.

The scenes contain isolated Lambertian objects on black backgrounds. Although the geometry is relatively simple, ten views leave substantial portions of the surface weakly observed or unobserved. This setting therefore retains ambiguity between geometry, reflectance, and visibility while allowing controlled evaluation of known-light supervision.

Reference images are rendered in Blender Cycles using the same direct point-light illumination model assumed during optimization, Lambertian materials, and no indirect illumination. Blender serves as an independent renderer and avoids coupling the dataset-generation implementation to our optimization renderer.

Initialization. For all scenes, we initialize every point-based method from the same set of 25 white primitives positioned on a uniform grid within the scene bounding volume. The primitives share identical initial positions, white appearance, scale 0.1, opacity 0.5, and vertical orientation. For our beta surfels, we additionally initialize the shape parameter as $\beta_k = 0$.

Baselines. Our primary comparisons are against the point-based methods 2D-GS [4], SuGaR [3], RadiosityGS [5], and GOF [21]. We additionally include NeuS [18] and NeuS2 [19], which use implicit neural surface representations, and

GeoSVR [9], which uses a sparse-voxel representation, to contextualize the point-based results within the broader landscape of surface reconstruction methods.

For 2DGS, we report the standard 30K-iteration configuration from the original method. We additionally report a 7K baseline which is the same iteration count that SuGaR uses for 3DGS before applying their meshes extraction procedure. For GeoSVR, we use the official implementation without its monocular depth constraint. Monocular depth predictions were unreliable for our black background object images without foreground masks and introduced incorrect geometric constraints. No foreground masks or external monocular depth priors are provided to the evaluated methods.

4.2 Implementation Details

All experiments use a custom physically based renderer implemented in modern C++ and accelerated with SYCL [16]. We tested the same implementation on AMD and NVIDIA GPUs using the corresponding SYCL backends. Optimization requires approximately 5–8 GB of VRAM and uses Adam [8] through PyTorch [14], with default optimizer settings except for parameter-specific learning rates.

We selected the regularization weights empirically in preliminary experiments and fixed them across all reported scenes avoiding per scene specific tuning: $\lambda_d = 100$, $\lambda_n = 0.005$, and $\lambda_{\text{weak}} = 0.05$ in Eq. (20). Our optimization runs for 60K iterations. We extract meshes by truncated signed distance function (TSDF) fusion of depth maps rendered from the optimized surfels at the input camera poses, following the depth-fusion procedure used by 2DGS [4]. Ground-truth geometry is not used during optimization or mesh extraction.

Evaluation. We report symmetric Chamfer distance between reconstructed and ground-truth meshes. Accuracy denotes reconstruction-to-ground-truth distance and penalizes spurious geometry, while completion denotes ground-truth-to-reconstruction distance and penalizes missing surface regions.

5 Results

Geometry Reconstruction Table 1 compares our method with implicit, sparse-voxel, and point-based reconstruction baselines. Our method achieves the lowest mean symmetric Chamfer distance and the best result on three scenes. Among the point-based methods, it performs best on four scenes, with SuGaR performing slightly better on *LEGO*.

NeuS2 performs poorly without foreground masks or monocular-depth supervision. NeuS is the strongest non-point-based baseline, performing slightly better on *LEGO* and *Plant*, while our method performs better on the remaining scenes. GeoSVR is less accurate on average, particularly on *Plant*, however, we omit its monocular-depth prior because its predictions were unreliable on these images.

Table 1: Main geometry reconstruction results for sparse-view object reconstruction. Entries report symmetric Chamfer distance (CD), scaled by 10^2 for readability; lower is better. The mean is computed across all five scenes. The final column reports the average number of optimized primitives for point-based methods. Within the point-based methods, the **best**, **second-best**, and **third-best** results are highlighted.

Method	Teapot	LEGO	Horse	Dragon	Plant	Mean ↓	Avg pts ↓
NeuS2 [19]	95.850	112.344	90.242	101.955	93.212	98.721	–
NeuS [18]	1.396	1.424	2.022	1.397	2.180	1.684	–
GeoSVR [9] †	3.667	1.446	2.054	3.255	10.296	4.144	–
2DGS 7K [4]	2.471	1.429	1.702	2.251	2.891	2.149	43.0k
2DGS 30K [4]	53.570	30.563	39.075	53.149	15.917	38.455	51.0k
SuGaR [3]	5.837	1.406	1.993	2.613	3.468	3.064	421.4k
GOF [21]	3.450	65.990	2.394	2.074	3.186	15.419	72.4k
RadiosityGS [5]	31.621	24.400	27.327	28.256	22.867	26.894	107.4k
Ours	1.387	1.426	1.319	1.297	2.252	1.536	267

† GeoSVR is evaluated without its monocular-depth prior; see Section 4.1.

Table 2: Unscaled mean directional Chamfer distance for the point-based methods across five scenes. Accuracy penalizes spurious geometry, while completion penalizes missing regions.

Method	Accuracy ↓	Completion ↓
2DGS 7K [4]	0.020	0.023
2DGS 30K [4]	0.743	0.026
SuGaR [3]	0.037	0.024
GOF [21]	0.295	0.014
RadiosityGS [5]	0.512	0.026
Ours	0.017	0.014

We use the standard 30K-iteration 2DGS schedule as the primary baseline configuration. We additionally report the 7K checkpoint to illustrate an optimization behavior observed in our sparse-view setting. Although the 30K schedule further reduces the image-space objective, later iterations continue to introduce primitives near object boundaries that help explain the black background and reduce photometric residuals. These additional primitives do not improve the extracted surface and instead lead to substantially worse mesh accuracy.

The 7K checkpoint therefore achieves lower Chamfer error in our evaluation and indicates that continued densification and improved image fitting do not necessarily yield better geometry. Figure 3 illustrates this distinction for the 2DGS-30K and 7K results. In contrast, our method is optimized for 60K iterations without the same late-stage degradation in extracted geometry.

RadiosityGS is the closest physics-based point-based baseline. In our setting, it exhibits a related mismatch between image-space optimization and extracted

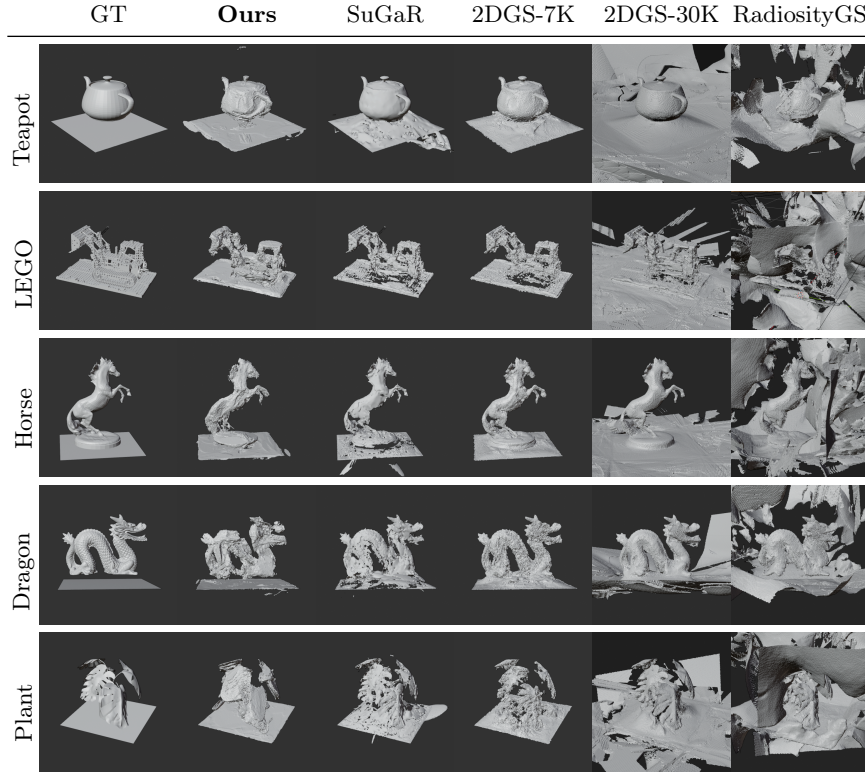


Fig. 3: Qualitative reconstruction comparison across synthetic object scenes. We compare the ground-truth geometry with our reconstruction and baseline methods. Our method preserves the main object structure while using fewer optimized primitives.

mesh quality similar to 2DGS-30k. This suggests that, as with 2DGS, continued primitive growth can improve image fitting without improving the recovered surface. Nevertheless, the Dragon reconstruction in Figure 3 shows a notable benefit. RadiosityGS and our method are the only evaluated approaches that separate the object from the background plane, whereas the remaining methods fuse the two surfaces.

The 2DGS-30K reconstructions can appear smoother in some examples, but visual smoothness does not consistently correspond to geometric fidelity. The directional Chamfer results in Table 2 show that our improvement is not obtained merely by omitting uncertain regions: our method improves accuracy while maintaining competitive completion. Table 1 further shows that this improvement is achieved using only 267 surfels on average, compared with tens to hundreds of thousands of primitives for the point-based baselines. This combination of lower geometric error and a substantially smaller representation suggests a favorable scaling direction where increasing scene complexity need not require the rapid primitive growth observed for the evaluated splatting-based approaches.

Table 3: Ablation study on the Horse scene using ten input views. $\mathcal{L}_d$: depth distortion loss; $\mathcal{L}_n$: normal consistency loss; $\mathcal{L}_\alpha$: weak opacity prior. CD denotes unscaled symmetric Chamfer distance; lower is better.

Setting	$\mathcal{L}_d$	$\mathcal{L}_n$	$\mathcal{L}_\alpha$	Learn β	CD $\downarrow$	Acc. $\downarrow$	Comp. $\downarrow$
Full	✓	✓	✓	✓	0.0132	0.0138	0.0125
w/o $\mathcal{L}_d$		✓	✓	✓	0.0162	0.0171	0.0152
w/o $\mathcal{L}_n$	✓		✓	✓	0.0185	0.0193	0.0178
w/o $\mathcal{L}_\alpha$	✓	✓		✓	0.0241	0.0187	0.0295
Fixed β	✓	✓	✓		0.0278	0.0245	0.0312

These results indicate accurate recovery of the overall object geometry and major surface structures rather than fine-scale detail, making the method more suitable for applications such as coarse path planning, collision-aware navigation, and compact scene representation than high-fidelity digitization.

Ablations Table 3 evaluates the main components of the method on the Horse scene. We include depth distortion and normal regularization to show that they also benefit our optimization similar to 2DGS [4]. The weak visibility-weighted opacity regularizer has a larger impact on the result than depth and normal regularization and shows that for our simple scenes reducing the opacity/radiance ambiguity is beneficial. On Horse, fixing the beta profile similarly degrades reconstruction quality, indicating that the chosen opacity kernel is beneficial over a Gaussian-like shape only. This isolates the benefit of adapting the local opacity falloff during optimization.

6 Conclusion

We presented a geometry first point-based reconstruction method based on compact opacity-bearing beta surfels and adjoint light transport with explicit transmittance. Across five synthetic objects reconstructed from ten posed views, the method achieves the lowest mean symmetric Chamfer distance among the evaluated baselines while using only 267 surfels on average. These results indicate that physically informed optimization can recover accurate coarse scale geometry without relying on dense primitive sets.

The present system is a deliberately constrained first instantiation and is not intended as the final form of the method. We evaluate isolated synthetic objects under known direct illumination and Lambertian reflectance, although the formulation admits recursive transport. Extending the system to indirect illumination, jointly estimated illumination, non-Lambertian reflectance, real captures, and more adaptive surfel refinement may provide stronger reconstruction cues. The current results therefore establish the viability of compact transport-optimized surfels while leaving substantial transport and representational capacity unexplored.

References

1. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: SIGGRAPH '96. pp. 303–312. ACM (1996). <https://doi.org/10.1145/237170.237269>
2. Do Carmo, M.P.: Differential Geometry of Curves and Surfaces. Dover Publications, revised and updated second edn. (2016)
3. Guédon, A., Lepetit, V.: SuGaR: Surface-aligned Gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. In: CVPR. pp. 5354–5363 (2024). <https://doi.org/10.1109/CVPR52733.2024.00512>
4. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2D Gaussian splatting for geometrically accurate radiance fields. In: SIGGRAPH 2024 Conference Papers. pp. 32:1–32:11. ACM (2024). <https://doi.org/10.1145/3641519.3657428>
5. Jiang, K., Sun, J.M., Li, Z., Wang, D., Li, T.M., Ramamoorthi, R.: Differentiable light transport with Gaussian surfels via adapted radiosity for efficient relighting and geometry reconstruction. ACM Transactions on Graphics **44**(6), 210:1–210:25 (2025). <https://doi.org/10.1145/3763305>
6. Kajiya, J.T.: The rendering equation. Computer Graphics (SIGGRAPH '86) **20**(4), 143–150 (1986). <https://doi.org/10.1145/15886.15902>
7. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D Gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4), 139:1–139:14 (2023). <https://doi.org/10.1145/3592433>
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of International Conference for Learning Representations (ICLR) (2015). <https://doi.org/10.48550/arXiv.1412.6980>
9. Li, J., Zhang, J., Zhang, Y., Bai, X., Zheng, J., Yu, X., Gu, L.: GeoSVR: Taming sparse voxels for geometrically accurate surface reconstruction. In: Advances in Neural Information Processing Systems. vol. 38, pp. 108809–108837. Curran Associates (2025)
10. Li, T.M., Aittala, M., Durand, F., Lehtinen, J.: Differentiable Monte Carlo ray tracing through edge sampling. ACM Transactions on Graphics **37**(6), 222:1–222:11 (2018). <https://doi.org/10.1145/3272127.3275109>
11. Liu, R., Sun, D., Chen, M., Wang, Y., Feng, A.: Deformable beta splatting. In: SIGGRAPH 2025 Conference Papers. pp. 101:1–101:11 (2025). <https://doi.org/10.1145/3721238.3730716>
12. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In: ECCV. Lecture Notes in Computer Science, vol. 12346. Springer (2020). https://doi.org/10.1007/978-3-030-58452-8_24
13. Nimier-David, M., Speierer, S., Ruiz, B., Jakob, W.: Radiative backpropagation: An adjoint method for lightning-fast differentiable rendering. ACM Transactions on Graphics **39**(4), 146:1–146:15 (2020). <https://doi.org/10.1145/3386569.3392406>
14. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 32. Curran Associates (2019)
15. Stam, J.: Computing light transport gradients using the adjoint method. arXiv:2006.15059 [cs.GR] (2020). <https://doi.org/10.48550/arXiv.2006.15059>

16. The Khronos SYCL Working Group: SYCL™ 2020 Specification. <https://registry.khronos.org/SYCL/specs/sycl-2020/pdf/sycl-2020.pdf> (2025), revision 11, accessed July 3, 2026
17. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: SIGGRAPH '94. pp. 311–318. ACM (1994). <https://doi.org/10.1145/192161.192241>
18. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 34, pp. 27171–27183. Curran Associates (2023)
19. Wang, Y., Han, Q., Habermann, M., Daniilidis, K., Theobalt, C., Liu, L.: NeuS2: Fast learning of neural implicit surfaces for multi-view reconstruction. In: ICCV. pp. 3272–3283. IEEE (2023). <https://doi.org/10.1109/ICCV51070.2023.00305>
20. Worchel, M., Finnendahl, U., Alexa, M.: Radiative backpropagation with non-static geometry. In: Eurographics Symposium on Rendering (EGSR). The Eurographics Association (2025). <https://doi.org/10.2312/sr.20251198>
21. Yu, Z., Sattler, T., Geiger, A.: Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics* **43**(6), 271:1–271:13 (2024). <https://doi.org/10.1145/3687937>
22. Yu, Z., Zhang, C., Nowrouzezahrai, D., Dong, Z., Zhao, S.: Efficient differentiation of pixel reconstruction filters for path-space differentiable rendering. *ACM Trans. Graph.* **41**(6) (Nov 2022). <https://doi.org/10.1145/3550454.3555500>, <https://doi.org/10.1145/3550454.3555500>
23. Zhang, C., Miller, B., Yan, K., Gkioulekas, I., Zhao, S.: Path-space differentiable rendering. *ACM Transactions on Graphics* **39**(4), 143:1–143:19 (2020). <https://doi.org/10.1145/3386569.3392383>
24. Zhang, C., Wu, L., Zheng, C., Gkioulekas, I., Ramamoorthi, R., Zhao, S.: A differential theory of radiative transfer. *ACM Transactions on Graphics* **38**(6), 227:1–227:16 (2019). <https://doi.org/10.1145/3355089.3356522>
25. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: EWA splatting. *IEEE Transactions on Visualization and Computer Graphics* **8**(3), 223–238 (2002). <https://doi.org/10.1109/TVCG.2002.1021576>