

PanoLess: Environment Reconstruction from Partial Reflective Views

Ahitagni Das¹, Ashok Veeraraghavan², and Vivek Boominathan¹

Rice University, Houston TX 77005, USA

Abstract. Reflections from shiny objects and glass façades naturally extend the field of view of a camera, capturing the surrounding environment without the need to pan the camera or acquire a full panorama. We propose PanoLess, a Gaussian-splat-based framework that reconstructs the surrounding environment as a distant-illumination map from images captured on only one side of a reflective surface. PanoLess leverages surface-aligned 2D Gaussian splats with deferred shading to recover accurate per-pixel normals and reflection cues, which are fused into a neural cubemap representation of the environment. In addition, PanoLess produces a visibility map that explicitly denotes which regions of the environment are supported by the partial reflective observations. Unlike existing inverse-rendering and reflection-aware Gaussian-splatting approaches—which typically require full 360° coverage and struggle under incomplete views—PanoLess enables consistent, physically grounded illumination estimation from partial-view input. We show that PanoLess achieves high-fidelity and geometrically consistent environment reconstruction, outperforming reflection-aware baselines on a new custom synthetic benchmark and publicly available datasets, and demonstrating generalization to real-world reflective captures.



Fig. 1: PanoLess recovers the surrounding environment from partial views of a reflective object—without panoramic capture. From images taken on one side of a shiny surface, PanoLess estimates an environment map alongside a visibility map that distinguishes well-observed regions from unsupported ones.

1 Introduction

Reflective surfaces such as glass façades, metals, and polished objects offer a unique opportunity to view parts of the environment that the camera itself cannot directly see. Unlike diffuse surfaces, which follow Lambertian assumptions commonly exploited in multi-view reconstruction [11, 28], specular reflections act

as indirect imaging pathways that naturally expand the camera’s field of view without requiring panning or full-panorama capture. However, the appearance of a reflective surface is jointly determined by its geometry and the surrounding illumination, creating an inherent ambiguity: different combinations of surface normals and lighting can reproduce the same observations [1, 25]. In practical scenarios, only one side of a reflective object is typically visible, leading to partial and incomplete reflection cues that encode information about both object shape and the unseen environment. These cues are difficult to interpret reliably, and both classical multi-view stereo [28] and recent neural rendering pipelines [30] struggle to disentangle geometry from illumination under such partial-view conditions.

Recovering environment maps from only partial reflective observations is a powerful means of expanding spatial awareness for robotics, AR/VR, and autonomous navigation. Yet, most existing inverse-rendering and reflection-aware methods implicitly assume dense, near-360° coverage or rely on strong supervision, making them brittle in realistic, limited-view scenarios.

Recent progress in point- and splat-based neural scene representations has opened new possibilities for real-time, high-quality rendering. Among these, Gaussian Splatting [10, 16] has emerged as a leading paradigm for efficient scene reconstruction and differentiable rendering. However, volumetric Gaussians lack explicit surfaces and therefore produce inconsistent normals and reflection directions, limiting their ability to reason about specular phenomena. Surface-aligned 2D Gaussian splats address this by explicitly modeling tangent-plane geometry and normals, but prior extensions still rely on simplified or low-frequency lighting models that are insufficient to reconstruct the complex, high-frequency structure of real-world environment illumination—especially when large portions of the environment remain unseen.

We introduce PanoLess¹, a framework that recovers the surrounding environment as a distant-illumination map (a 2D cubemap) from images taken on only one side of a reflective surface—without requiring large camera-panning or panoramic capture. PanoLess builds on the accurate normals and reflection cues provided by 2D Gaussian Splatting with deferred shading and aggregates them into a neural cubemap representation capable of modeling high-frequency directional radiance. Crucially, we also introduce a visibility map that quantifies which regions of the environment are directly supported by available reflections versus those that must be inferred. This explicit reasoning about missing environment regions enables consistent, physically grounded illumination estimation under highly incomplete observability. Our contributions are as follows:

- We formulate *partial-view* environment reconstruction: recovering high-fidelity environment map from reflections observed on only one side of a specular surface, without panoramic capture.
- We introduce a *visibility map*, a per-pixel confidence of the environment map, based on the rays sampled by the reflective object geometry.

¹ Project Page: <https://vb-gee.github.io/panoless/>

- We introduce the Shiny Partial benchmark for partial-view environment reconstruction with ground-truth environment maps, showing PanoLess achieves nearly 5 dB improvement over baselines with sub-5° mean normal error.
- We validate PanoLess on real-world handheld captures, showing it generalizes beyond synthetic data without domain-specific supervision.

2 Related Work

3D Gaussian Splatting and Extensions. 3D Gaussian Splatting (3DGS) [16] represents a scene with anisotropic volumetric Gaussians rasterized in real time, achieving unprecedented speed and quality for novel view synthesis. However, the original 3DGS encodes view-dependence using low-order spherical harmonics [25], which cannot represent sharp specular highlights. More importantly, 3D Gaussians lack explicit surfaces, leading to ambiguous surface normals and inconsistent reflections. Several extensions attempt to address these limitations. GaussianShader [14] introduces heuristic normal estimation from Gaussian covariances to enable shading, though normals remain unstable. Relightable 3D Gaussians [7] factor appearance into diffuse and specular components with estimated normals and materials, supporting relighting. GS-IR [19] performs full inverse rendering of geometry, BRDF, and lighting within a Gaussian framework, while 3DGS-DR [36] introduces deferred reflection rendering via screen-space ray tracing. Spec-Gaussian [34] replaces spherical harmonics with anisotropic spherical Gaussians to model high-frequency lobes more faithfully. DN-Splatter [29] incorporates monocular depth and normal priors to regularize geometry. Despite these advances, all 3DGS-based approaches inherit the fundamental limitation of volumetric splats: without true surfaces, they cannot guarantee consistent normals and reflection directions, often leading to blurred or incorrect highlights.

2D Gaussian Splatting and Reflective Surfaces. 2D Gaussian Splatting (2DGS) [10] overcomes the geometric ambiguity of 3DGS by introducing surface-aligned primitives: each Gaussian is a planar surfel defined by two tangent vectors, yielding explicit and consistent normals. This representation significantly improves reconstruction accuracy on standard datasets [12]. Extensions of 2DGS include PGSR [3] (combining 2DGS with neural SDFs), GaussianSurfels [4], Gaussian Opacity Fields [37], and Street Gaussians [33], all of which emphasize geometry fidelity. For reflective surfaces, early work such as Mirror-3DGS [21] handles planar mirrors by ray-marching, while GaussianRoom [31] models indoor reflective scenes but relies on low-frequency lighting. Most recently, Ref-GS (CVPR 2025) [39] introduces deferred shading for specular effects on 2DGS, but its lighting is implicit in the shading model. Overall, 2DGS offers a geometrically sound foundation for reflections, but current methods lack expressive environment lighting. A parallel line of reflection-aware methods targets specular appearance through explicit lighting or material models—via neural or directional environment lighting [13, 20], physically-based deferred rendering on 2DGS [35], environment Gaussians and 2D Gaussian ray tracing [8, 32], or multi-view material priors [38]; orthogonally, generative priors hallucinate plausible illumination from a single image [24].

Environment Maps and Joint Illumination. Environment maps are a cornerstone of rendering reflective materials [2,5,6], enabling high-quality reflections when surface normals are known. Classical models based on spherical harmonics [15,26] capture only low-frequency illumination, limiting realism. Neural approaches such as Neural Radiosity [9], EnvMapNet [27], and ENVIDR [18] introduce learnable environment representations that capture high-frequency lighting. Joint optimization of geometry and illumination has been shown to improve reconstruction accuracy [17,22,23]. In contrast, prior works rely on volumetric or implicit scene models, whereas our method combines surface-aligned 2DGS with neural environment maps for high-fidelity, real-time reflective reconstruction.

3 Methodology

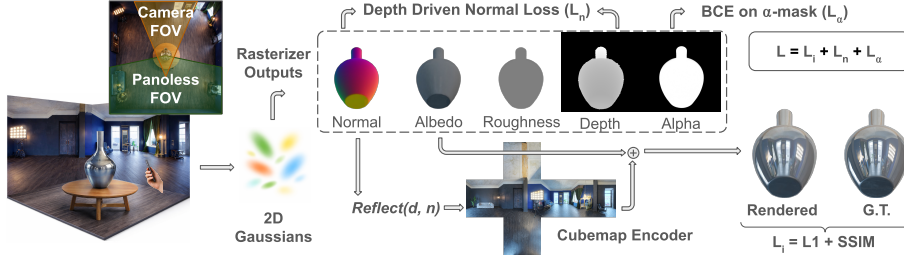


Fig. 2: Overview of our rendering pipeline. Given a set of optimizable 2D Gaussians, we rasterize per-Gaussian properties to obtain normal, albedo, roughness, depth, and alpha maps. Surface normals and view directions give reflection directions, which query a learnable cubemap encoder to produce the specular color; this is composed with the diffuse albedo to yield the final render.

Our pipeline consists of three stages: (i) a Gaussian splatting pass that rasterizes surface-aligned primitives into screen-space albedo and geometry buffers, (ii) a deferred shading pass that evaluates specular reflection per pixel using a learned neural cubemap, and (iii) a visibility accumulation pass that tracks which directions of the environment are supported by observations. An overview of our method is illustrated in Fig. 2.

3.1 Scene Representation

We represent the scene as a collection of K surface-aligned 2D Gaussian primitives $\{G_i\}_{i=1}^K$, following Huang *et al.* [10]. Each primitive G_i is a planar surfel defined by a center $\mu_i \in \mathbb{R}^3$, two tangent vectors $\mathbf{t}_{u,i}, \mathbf{t}_{v,i} \in \mathbb{R}^3$ that span a local tangent plane, an opacity $o_i \in [0, 1]$, and a diffuse linear RGB color $\mathbf{c}_i^d \in \mathbb{R}^3$. The surface normal is given explicitly by $\mathbf{n}_i = \mathbf{t}_{u,i} \times \mathbf{t}_{v,i} / \|\mathbf{t}_{u,i} \times \mathbf{t}_{v,i}\|$. This construction guarantees that every Gaussian carries a well-defined, geometrically consistent normal, in contrast to volumetric 3D Gaussians whose shortest-axis heuristic produces unstable orientations [16].

3.2 Deferred Rasterization

Given a camera with view matrix $\mathbf{V}$ and projection matrix $\mathbf{P}$, we project the 2D Gaussians onto the image plane and perform front-to-back alpha compositing. For a generic per-primitive attribute ψ_i , the composited screen-space buffer at pixel $\mathbf{p}$ is

$$\Psi(\mathbf{p}) = \sum_{i \in \mathcal{N}} \psi_i o_i \mathcal{G}_i(\mathbf{p}) \prod_{j < i} (1 - o_j \mathcal{G}_j(\mathbf{p})), \quad (1)$$

where $\mathcal{N}$ denotes the depth-sorted set of primitives overlapping $\mathbf{p}$, o_i is the per-primitive opacity, and $\mathcal{G}_i(\mathbf{p})$ is the screen-space Gaussian evaluation. By choosing ψ_i as $\mathbf{c}_i^d$, we obtain the diffuse albedo map $\mathbf{A}$. In addition to the Albedo, the rasterizer outputs a seven-channel auxiliary map

$$\mathbf{M} = [D_{\text{exp}}, \alpha, \mathbf{N}_{\text{rast}}, D_{\text{med}}, D_{\text{dist}}] \in \mathbb{R}^{7 \times H \times W}, \quad (2)$$

where D_{exp} and D_{med} are the expected and median depth, α is the accumulated opacity, $\mathbf{N}_{\text{rast}} \in \mathbb{R}^3$ is the rasterized normal in camera coordinates, and D_{dist} encodes depth distortion for regularization [10].

3.3 Surface Geometry Extraction

Robust surface geometry is critical for computing accurate reflection directions. We derive a composite surface depth by blending the expected and median depth maps:

$$D_{\text{surf}}(\mathbf{p}) = (1 - \lambda_d) D_{\text{exp}}(\mathbf{p}) + \lambda_d D_{\text{med}}(\mathbf{p}), \quad (3)$$

with $\lambda_d = 0.5$. The blending mitigates noise in expected depth caused by semi-transparent regions while retaining the sharp discontinuities captured by the median. Surface normals are then recovered by differentiating the depth map in screen space:

$$\mathbf{n}_{\text{surf}}(\mathbf{p}) = \text{normalize} \left(\frac{\partial D_{\text{surf}}}{\partial u} \times \frac{\partial D_{\text{surf}}}{\partial v} \right), \quad (4)$$

where u, v denote image-plane coordinates. These depth-derived normals complement the rasterized normals $\mathbf{N}_{\text{rast}}$ from the alpha-composited primitive orientations; we transform the latter from camera to world space via $\mathbf{N}_{\text{world}} = \mathbf{V}_{3 \times 3}^\top \mathbf{N}_{\text{rast}}$ and re-normalize. In practice, we use $\mathbf{N}_{\text{world}}$ for the reflection computation and supervise $\mathbf{n}_{\text{surf}}$ as a regularizer.

3.4 Deferred Specular Shading

We model the environment as distant illumination: radiance depends only on direction, so the surroundings form a function on $\mathbb{S}^2$ stored as a 2D cubemap. PanoLess thus recovers an environment map, with the Gaussian splats modeling only the reflective surface. This far-field assumption is standard in inverse rendering and accurate when the reflector is small relative to its distance from the scene, as for the façades and outdoor objects we target. Our shading model operates entirely in screen space, after the rasterization pass has produced the albedo and geometry buffers.

Reflection direction. Let $\mathbf{v}(\mathbf{p})$ denote the unit camera ray direction at pixel $\mathbf{p}$, obtained from the camera intrinsics and pixel coordinates. The specular reflection direction with respect to the surface normal $\mathbf{n}(\mathbf{p})$ is

$$\boldsymbol{\omega}_r(\mathbf{p}) = \text{normalize}(\mathbf{v}(\mathbf{p}) - 2(\mathbf{v}(\mathbf{p}) \cdot \mathbf{n}(\mathbf{p}))\mathbf{n}(\mathbf{p})). \quad (5)$$

Sensitivity to normal error. Since each cubemap query is indexed by $\boldsymbol{\omega}_r$, the fidelity of the recovered environment is governed by the accuracy of the surface normal. Perturbing the normal $\mathbf{n} \rightarrow \mathbf{n} + \delta\mathbf{n}$ at fixed view direction and dropping second-order terms, the reflection equation gives

$$\delta\boldsymbol{\omega}_r = -2[(\mathbf{v} \cdot \mathbf{n})\delta\mathbf{n} + (\mathbf{v} \cdot \delta\mathbf{n})\mathbf{n}], \quad \|\delta\boldsymbol{\omega}_r\| \leq 2(|\mathbf{v} \cdot \mathbf{n}| + 1)\|\delta\mathbf{n}\| \leq 4\|\delta\mathbf{n}\|. \quad (6)$$

The reflection-direction error is thus first-order linear in the normal error, with the largest amplification at grazing angles. Because pose error propagates directly into the estimated normals, this predicts a correspondingly sharp sensitivity of the recovered environment to camera-pose accuracy, which we quantify in Sec. 4.5.

Neural cubemap. We represent the environment map as a learnable cubemap $\mathcal{E} : \mathbb{S}^2 \rightarrow \mathbb{R}^3$, stored as six trainable $L \times L$ RGB grids—one per cube face—collected in a single $6 \times 3 \times L \times L$ parameter tensor. For a query direction $\boldsymbol{\omega} \in \mathbb{S}^2$, given directly in Cartesian coordinates, we determine the dominant face, compute the (s, t) parameterization, and retrieve the radiance via bilinear interpolation on the corresponding face grid:

$$\mathbf{L}_{\text{env}}(\boldsymbol{\omega}) = \sigma(\mathcal{E}(\boldsymbol{\omega})), \quad (7)$$

where $\sigma(\cdot)$ is the sigmoid activation, constraining the output to $[0, 1]$. The reflection direction is fed straight to the sampler, so the specular color is exactly $\sigma(\mathcal{E}(\boldsymbol{\omega}_r))$ with no network in between. The environment map is optimized jointly with the Gaussian parameters through the photometric loss, with gradients flowing through the bilinear sampler. Unlike methods relying on spherical harmonics [16] or pre-filtered mip-mapped cubemaps [38], our single-level cubemap directly stores high-frequency radiance, sufficient for mirror-like reflections. Ref-GS-style encoders [38] select a mip level from per-pixel roughness and decode it with an MLP; for our near-zero-roughness targets this collapses to the sharpest mip level, adding parameters without changing the output. A direct lookup gives the photometric loss an unattenuated gradient path to every texel; our ablation (Sec. 4.4) shows reintroducing roughness attenuation degrades both the render and the recovered environment—invisible under dense capture, but decisive under partial views where attenuation starves supervision exactly where samples are sparse.

Color composition. We compose the final per-pixel color as the sum of a diffuse and a specular component:

$$\mathbf{C}(\mathbf{p}) = \underbrace{\mathbf{A}(\mathbf{p})}_{\text{diffuse}} + \underbrace{\mathbf{L}_{\text{env}}(\boldsymbol{\omega}_r(\mathbf{p}))}_{\text{specular}}, \quad (8)$$

where $\mathbf{A}$ is the rasterized albedo map. Notably, the specular term is taken directly from the environment map without an intermediate MLP or learned blending weight. This stands in contrast to prior work that modulates the environment contribution by a roughness-dependent reflection strength [36] or evaluates a full BRDF integral [38]. For the highly reflective surfaces we target, a learned blending weight empirically converges to unity, so the direct connection yields cleaner gradients and faster convergence without the redundant parameterization.

3.5 Training Objective and Protocol

We optimize all learnable parameters—Gaussian attributes $\{G_i\}$ and the environment map $\mathcal{E}$ —end-to-end with a photometric loss. Following standard practice in Gaussian splatting [16, 36], we combine an ℓ_1 term with a structural similarity term:

$$\mathcal{L}_{\text{color}} = (1 - \lambda_s) \mathcal{L}_1 + \lambda_s \mathcal{L}_{\text{D-SSIM}}, \quad (9)$$

where $\lambda_s = 0.2$. The individual terms are evaluated on the composited image:

$$\mathcal{L}_1 = \|\mathbf{C}_{\text{out}} \odot \alpha + (1 - \alpha) \odot \mathbf{b} - \mathbf{I}_{\text{gt}}\|_1, \quad (10)$$

$$\mathcal{L}_{\text{D-SSIM}} = 1 - \text{SSIM}(\mathbf{C}_{\text{out}} \odot \alpha + (1 - \alpha) \odot \mathbf{b}, \mathbf{I}_{\text{gt}}), \quad (11)$$

where $\mathbf{b}$ is the background color and $\mathbf{I}_{\text{gt}}$ the ground-truth image.

We incorporate a depth-normal consistency regularizer that encourages agreement between the rasterized normals and the depth-derived normals:

$$\mathcal{L}_{\text{n-d}} = \|1 - \mathbf{N}_{\text{world}}^\top \mathbf{n}_{\text{surf}}\|_1, \quad (12)$$

introduced by Huang *et al.* [10], which jointly sharpens the geometry and stabilizes the normal estimates that drive the reflection computation. During the first 3000 iterations, we additionally supervise the accumulated opacity with a binary cross-entropy loss against the ground-truth object mask:

$$\mathcal{L}_\alpha = \text{BCE}(\alpha(\mathbf{p}), m_{\text{gt}}(\mathbf{p})), \quad (13)$$

where m_{gt} is the ground-truth alpha mask. This early-stage supervision encourages the Gaussians to quickly converge to the object silhouette, substituting for the geometric anchoring that full-orbit multi-view consistency normally provides but which is unavailable when all cameras lie on one side. The total objective is

$$\mathcal{L} = \mathcal{L}_{\text{color}} + \lambda_{\text{nd}} \mathcal{L}_{\text{n-d}} + [t < 3000] \mathcal{L}_\alpha. \quad (14)$$

All Gaussian parameters and the environment map $\mathcal{E}$ are optimized jointly from iteration zero. Throughout training, we apply the adaptive density control of the base 2DGS [10], including periodic opacity reset and clone/split operations. Normal propagation [36] is employed to enlarge Gaussians, spreading well-estimated normals to neighboring primitives and accelerating convergence on reflective regions. The visibility map $\mathcal{V}$ is updated in an online fashion at each iteration with negligible overhead.

3.6 Visibility Accumulation

A partial-view capture of a reflective object only illuminates a subset of directions in the environment. Directions that are never queried during training have no photometric supervision and must be treated differently from well-observed regions. We therefore maintain a *visibility map* $\mathcal{V} : \mathbb{S}^2 \rightarrow [0, 1]$ on the same cube-map grid as $\mathcal{E}$, recording the cumulative evidence for each environment direction.

During training, for every pixel $\mathbf{p}$ with $\alpha(\mathbf{p}) > \tau$, we scatter an increment to $\mathcal{V}$ at the corresponding reflection direction $\omega_r(\mathbf{p})$, weighted by the pixel’s accumulated opacity. After training, $\mathcal{V}$ provides an explicit per-texel confidence score: high values indicate directions that are well-constrained by observed reflections, while near-zero values flag regions that were never reflected into the training views. This map serves two purposes: (i) it identifies the angular extent of the recovered environment, enabling downstream applications to distinguish trustworthy illumination from extrapolated content, and (ii) it can be used to modulate environment map regularization, applying stronger priors in unobserved regions. This signal is meaningful only under partial coverage: with dense 360° capture it saturates toward a constant, carrying information only when large portions of the environment go unsupported.

4 Experiments

Datasets. Existing reflection benchmarks [30] provide dense 360° coverage of objects with moderate specularity and no ground-truth environment maps, precluding evaluation of partial-view environment reconstruction on highly specular surfaces. We address this gap with two benchmarks. **Shiny Partial** is a Blender-based pipeline that renders images from camera arcs confined to a single hemisphere and exports the ground-truth environment map; we release the generator alongside three fixed scenes—*Cola*, *Vase*, and *Mirror*—with 100 training and 200 test images each. We also construct **Partial Shiny Blender** by restricting the Shiny Blender dataset [30] to a single-hemisphere subset, providing a mixed-specularity testbed discussed in Sec. 4.5. Additionally, we validate on real-world captures (**Shiny Real**) and present ablations in Sec. 4.4.

4.1 Shiny Partial Results

Table 1 reports quantitative results on our Shiny Partial benchmark. PanoLess outperforms all baselines on PSNR, SSIM, and L1 across every scene, and achieves the lowest LPIPS on two of three. On average, PanoLess attains 13.69 dB—a gain of nearly 5 dB over the next-best method (MaterialRefGS, 8.89 dB). The improvement is most pronounced on *Mirror* (+4.8 dB), where the planar geometry requires globally consistent normals to correctly resolve the reflected environment, a setting in which volumetric baselines fail.

Accurate surface normals are the single most critical factor for environment reconstruction: each pixel’s reflection direction is computed directly from the

Table 1: Quantitative results on Shiny Partial. We report environment map rendering quality (PSNR, SSIM, LPIPS, L1) and normal estimation accuracy (MAE, Median, RMSE in degrees, and percentage of pixels within 5° , 11.25° , and 22.5° thresholds). PanoLess achieves the best rendering metrics on every scene and produces substantially more accurate normals, with nearly 90% of pixels within 5° of the ground truth on average. **Best** and **second best** highlighted.

Scene	Method	Environment Map				Normal Estimation					
		PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	L1 $\downarrow$	MAE $\downarrow$	Median $\downarrow$	RMSE $\downarrow$	$< 5^\circ \uparrow$	$< 11.25^\circ \uparrow$	$< 22.5^\circ \uparrow$
Vase	3DGS-DR [36]	7.96	0.204	0.665	0.346	14.7644	5.0592	26.7193	49.82	63.24	78.73
	GShader [14]	9.62	0.121	0.668	0.264	28.4489	22.7388	34.4282	0.08	9.08	49.10
	Ref-Gaussian [35]	13.63	0.397	0.535	0.150	23.1096	17.1547	28.8376	1.03	27.43	64.48
	MaterialRefGS [38]	14.05	0.474	0.461	0.155	10.6664	6.2316	16.3371	30.39	83.10	91.87
	Ours	18.51	0.635	0.487	0.082	3.9683	2.7271	7.2938	88.10	96.40	98.87
Cola	3DGS-DR [36]	4.52	0.270	0.700	0.526	15.4632	4.4904	32.4032	52.16	74.12	85.03
	GShader [14]	6.47	0.118	0.736	0.402	33.5567	23.7507	43.7941	0.22	18.70	48.23
	Ref-Gaussian [35]	7.26	0.249	0.660	0.372	22.3515	16.9067	28.5207	3.47	27.32	65.79
	MaterialRefGS [38]	6.81	0.302	0.587	0.416	10.2303	8.0556	13.6171	21.55	79.39	93.55
	Ours	9.26	0.442	0.477	0.325	6.2487	3.2135	10.3550	82.98	87.48	91.28
Mirror	3DGS-DR [36]	6.22	0.274	0.711	0.420	46.5904	68.1785	52.7519	0.11	2.30	40.39
	GShader [14]	8.54	0.145	0.710	0.297	52.1474	52.1432	53.6931	0.00	0.00	2.25
	Ref-Gaussian [35]	5.56	0.226	0.746	0.457	14.7595	14.1869	15.2826	0.00	41.67	93.79
	MaterialRefGS [38]	5.82	0.291	0.755	0.449	14.0368	13.4597	15.0517	0.00	50.75	94.86
	Ours	13.31	0.337	0.655	0.177	4.6389	4.6243	4.6456	97.00	97.00	97.00
Avg.	3DGS-DR [36]	6.23	0.249	0.692	0.431	25.61	25.91	37.29	34.03	46.55	68.05
	GShader [14]	8.21	0.128	0.705	0.321	38.05	32.88	43.97	0.10	9.26	33.19
	Ref-Gaussian [35]	8.82	0.291	0.647	0.326	20.07	16.08	24.21	1.50	32.14	74.69
	MaterialRefGS [38]	8.89	0.356	0.601	0.340	11.64	9.25	15.00	17.31	71.08	93.43
	Ours	13.69	0.471	0.540	0.195	4.95	3.52	7.43	89.36	93.63	95.72

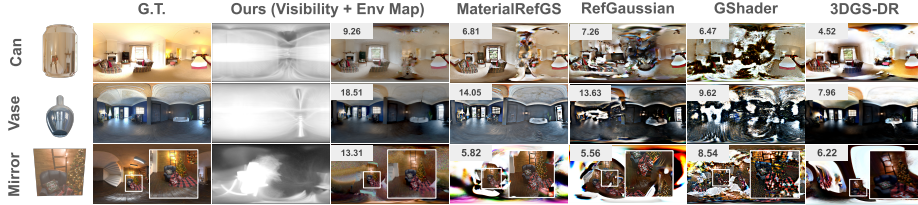


Fig. 3: Recovered environment maps on **Shiny Partial**. For each scene we show the ground-truth environment map, our reconstruction, our visibility map (bright $\rightarrow$ well-observed), and the environment maps extracted from each baseline. The PSNR against the G.T. environment map is mentioned for each on the top left.

estimated normal, so even small angular errors redirect the cubemap query and corrupt the recovered illumination. Table 1 reports normal estimation metrics alongside rendering quality, and Fig. 4 visualizes per-pixel angular error across all three scenes. PanoLess achieves a mean angular error of 4.95° on average, compared to 11.64° for the next-best method (MaterialRefGS), and places nearly 90% of pixels within 5° of the ground truth. The improvement is especially striking on *Mirror*, where PanoLess achieves 4.64° MAE with 97% of pixels below 5° , while all baselines exceed 14° MAE. This is because planar surfaces demand globally consistent normal orientation—any local perturbation produces

Table 2: Matched silhouette supervision. We retrain MaterialRefGS, our strongest baseline, with our identical $\mathcal{L}_\alpha$ applied for the first 3k iterations. Even with this advantage, it trails PanoLess on every scene and metric—most starkly on *Mirror*, where PanoLess places 97% of normals within 5° versus 0%. Normal estimation on Shiny Partial; **best** highlighted.

Scene	MaterialRefGS + $\mathcal{L}_\alpha$			PanoLess (Ours)		
	MAE ↓	Median ↓	$< 5^\circ \uparrow$	MAE ↓	Median ↓	$< 5^\circ \uparrow$
<i>Cola</i>	10.42	8.43	20.0	6.25	3.21	83.0
<i>Vase</i>	9.73	6.09	33.4	3.97	2.73	88.1
<i>Mirror</i>	14.79	13.99	0.0	4.64	4.62	97.0

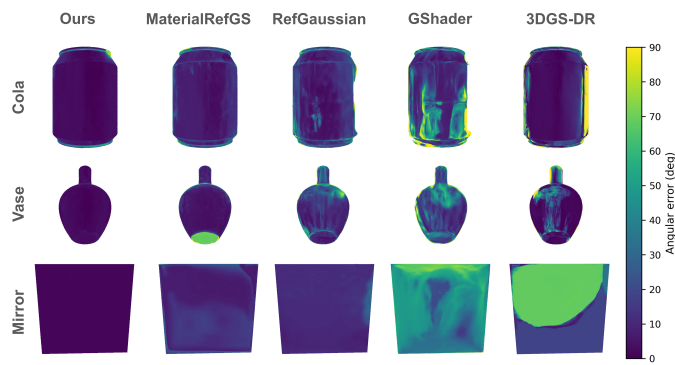


Fig. 4: Per-pixel normal angular error on Shiny Partial. We visualize the angular error (in degrees) between predicted and ground-truth surface normals for each method across all three scenes. PanoLess produces uniformly low error across the entire surface, while baselines exhibit large errors. The color scale ranges from 0° (dark) to 90° (bright).

a visibly incorrect reflection direction. Volumetric baselines lack the surface-aligned geometry to enforce this consistency, leading to the fragmented error maps visible in Fig. 4. On curved objects (*Cola*, *Vase*), PanoLess maintains smooth error fields that follow the surface curvature, whereas baselines show elevated error at silhouettes and high-curvature regions where the shortest-axis normal heuristic breaks down. These results validate our core design choice of building on 2D Gaussian surfels: the explicit tangent-plane normals provide the geometric foundation without which no amount of cubemap expressiveness can recover a coherent environment.

Fairness under matched silhouette supervision. A natural concern is whether PanoLess benefits unfairly from its early silhouette loss $\mathcal{L}_\alpha$. To isolate this factor, we retrain MaterialRefGS—our strongest baseline—with the identical $\mathcal{L}_\alpha$ applied for the same first 3k iterations (Tab. 2). The added supervision does improve its normal consistency over its original formulation, confirming that silhouette regularization is beneficial under sparse coverage; yet MaterialRefGS

still trails PanoLess on every scene and metric. The gap is widest on *Mirror*, where PanoLess places 97% of normals within 5° of ground truth against 0% for MaterialRefGS+ $\mathcal{L}_\alpha$. The advantage of PanoLess therefore stems from its surface-aligned representation rather than from silhouette supervision alone.

The fraction of the environment that can be recovered is governed by the reflector’s geometry. Curved surfaces such as *Cola* and *Vase* act as wide-angle mirrors: a small displacement in viewpoint sweeps the reflection direction across a large angular range, so even a single-hemisphere capture illuminates a substantial portion of the surrounding environment. A planar reflector such as *Mirror*, by contrast, maps each viewpoint to a narrow directional cone, and the cumulative angular coverage from the same capture configuration remains limited. The visibility maps in Fig. 3 corroborate this analysis: *Cola* and *Vase* exhibit broad high-confidence regions, whereas *Mirror* shows a compact observed band with the majority of the cubemap unsupported. Despite this restricted observability, PanoLess recovers coherent illumination within the supported region, while all baselines produce incoherent environment maps irrespective of reflector curvature.

Fig. 3 presents the recovered environment maps. PanoLess reconstructs recognizable illumination—room structure, light sources, and color gradients are faithfully preserved—and the accompanying visibility map explicitly delineates which directions are constrained by observed reflections versus extrapolated, an output unique to our method. All baselines yield severely degraded environments, ranging from washed-out low-frequency approximations (MaterialRefGS) to entirely incoherent reconstructions (3DGS-DR, GShader). These methods assume dense, near-360° coverage with moderate specularity and lack the direct cubemap supervision pathway exploited by PanoLess.

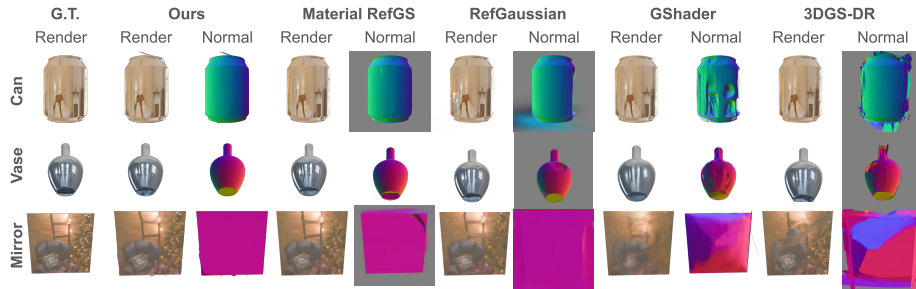


Fig. 5: Novel-view renders and surface normals on **Shiny Partial**.

Fig. 5 compares novel-view renders and surface normals. PanoLess produces smooth, geometrically accurate normal fields—clean gradients on the curved *Cola* and *Vase*, and a uniform orientation on the planar *Mirror*—closely matching the ground truth. Baselines exhibit noisy, fragmented, or erroneous normals that propagate into reflection artifacts and degraded render quality. RefGaussian and 3DGS-DR display particularly severe normal corruption on *Mir-*

ror, confirming that volumetric Gaussian representations cannot reliably model planar specular surfaces under partial-view conditions.

4.2 Downstream Use of the Visibility Map

The visibility map is useful on its own: for every direction, it indicates whether the recovered illumination is backed by real observations or not. A downstream application can act on this directly, trusting the well-observed directions and treating the rest with caution without any further processing. This makes the map a natural control signal for tasks such as gating generative inpainting, applying priors only where evidence is missing, or masking unreliable directions during relighting. The Gaussian blur described next is just one simple example of using the signal, not the point of it.

As a concrete illustration, thresholding $\mathcal{V}$ (Otsu) and feathering a Gaussian blur over the flagged low-confidence directions improves *every* environment-map metric on *Cola* (S4). The gain is small by design—blurring only the directions the map flags as unobserved never hurts—confirming the map is well-calibrated rather than being a contribution in its own right.

4.3 Shiny Real Results

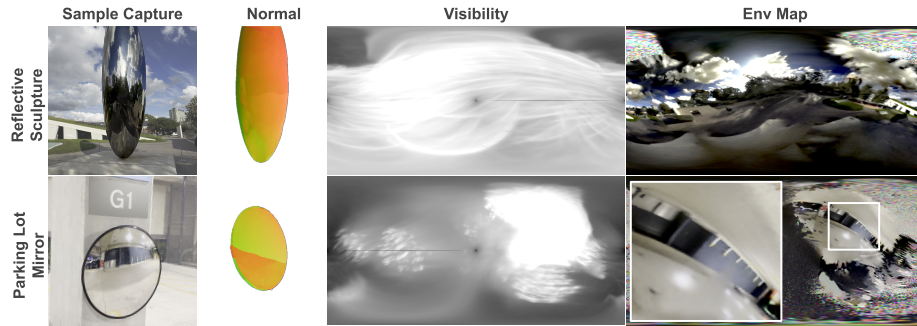


Fig. 6: Environment reconstruction on Shiny Real. Estimated normals, visibility maps, and recovered environment maps (with PSNR/SSIM/LPIPS/L1) for a chrome column and a parking-lot mirror.

We capture two scenes—a reflective sculpture and a parking-lot convex mirror—with handheld video from a single side of each reflector, extracting frames and poses via COLMAP and segmenting each object with a mask. As shown in Fig. 6, PanoLess recovers recognizable illumination—sky gradients, nearby buildings, overhead lights—directly from the partial reflections without synthetic supervision, and the visibility maps correctly highlight the regions constrained by the reflector’s curvature and camera trajectory.

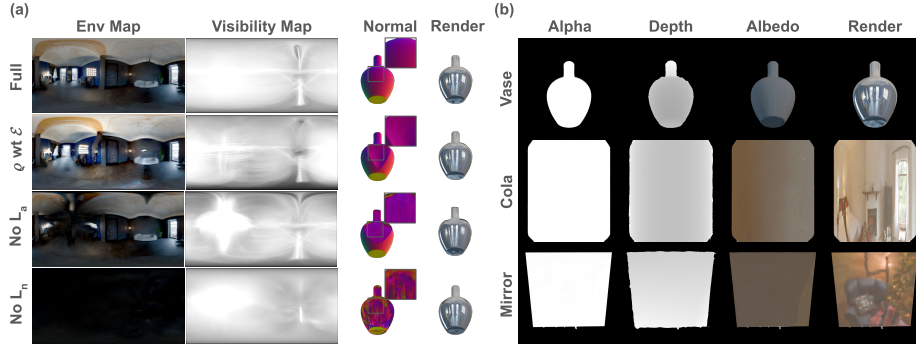


Fig. 7: (a) Ablation study on the *Vase* scene. Each row removes or modifies one component of the full model: ρ -weighted $\mathcal{E}$ scales the cubemap contribution by $(1-\rho)^2$ rather than using it directly, w/o $\mathcal{L}_\alpha$ removes early silhouette supervision, and w/o $\mathcal{L}_n$ removes normal consistency. We show the recovered environment map, visibility map, surface normals, and a novel-view render for each variant. (b) Output decomposition of the full model on all three Shiny Partial scenes, showing the alpha mask, depth, albedo, and final render.

Table 3: Ablation study on the *Vase* scene. We evaluate the impact of each component on novel view synthesis (Render) and environment map reconstruction (Env. Map). **Best** and **second best** highlighted.

Variant	Env. Map				Render			
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	L1 $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	L1 $\downarrow$
Full (Ours)	18.51	0.635	0.487	0.082	38.56	0.988	0.022	0.002
ρ -weighted $\mathcal{E}$	13.94	0.569	0.546	0.146	36.16	0.982	0.036	0.003
w/o $\mathcal{L}_\alpha$	12.91	0.466	0.559	0.177	33.75	0.977	0.045	0.004
w/o $\mathcal{L}_n$	8.68	0.098	0.621	0.337	35.82	0.978	0.042	0.004

4.4 Ablation and Decomposition

We ablate the key design choices of PanoLess on the *Vase* scene, evaluating both novel-view synthesis and environment map reconstruction (Table 3, Fig. 7).

Normal consistency ($\mathcal{L}_n$). Removing the depth-normal consistency loss produces the most severe degradation: environment-map PSNR drops from 18.51 to 8.68 dB and SSIM collapses to 0.098. Without this regularizer, surface normals become noisy and spatially incoherent (Fig. 7a, bottom row), which corrupts the reflection directions queried against the cubemap. The environment map receives conflicting gradients from incorrect directions and fails to converge to a meaningful solution. This confirms that accurate geometry is the single most critical prerequisite for environment recovery—the cubemap cannot compensate for broken normals.

Silhouette supervision ($\mathcal{L}_\alpha$). Removing the early binary cross-entropy loss costs 5.6 dB in environment-map PSNR and 4.8 dB in render PSNR. Without explicit silhouette guidance during the first 3000 iterations, Gaussians are slow to consolidate at the object boundary, producing alpha leakage that contaminates

normals along silhouette edges and injects erroneous reflection directions into the cubemap. The effect is visible in Fig. 7a as a noisier environment map and degraded normal quality near the object outline.

Direct cubemap supervision (ρ -weighted $\mathcal{E}$). This variant reintroduces roughness-dependent attenuation by scaling the cubemap contribution by $(1 - \rho)^2$, as is standard in prior work. Since the target surfaces are highly specular, roughness values are near zero but not exactly zero, which attenuates the cubemap signal and weakens the gradient pathway from the photometric loss to the environment map. The result is a 4.6 dB drop in environment-map PSNR and visible artifacts in the recovered environment (Fig. 7a, second row). Setting reflection strength to unity, as our full model does, provides the cubemap with an unattenuated gradient path. The roughness parameterization is not merely redundant for highly specular objects—it is actively harmful, as even small attenuation starves the cubemap of supervision.

Output decomposition. Fig. 7b visualizes the intermediate outputs of the full model. The alpha masks are near-binary and depth maps smooth, confirming high-quality surfel geometry. Notably, the albedo maps are low-energy and nearly uniform—the Gaussians attribute the dominant appearance to the cubemap rather than the diffuse channel, validating the additive shading model of Eq. 8.

4.5 Limitations

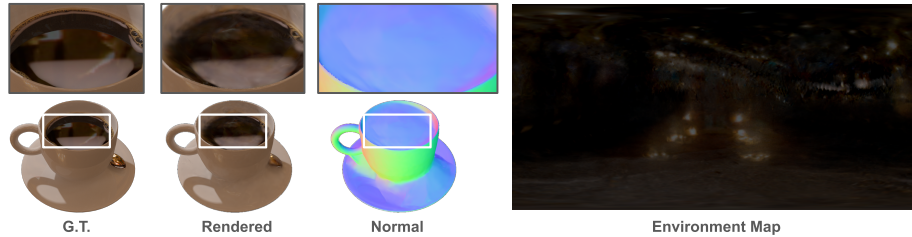


Fig. 8: Failure case. On a coffee cup scene, the coffee surface lacks metallic reflections, providing no signal for environment reconstruction. Our method is best suited for highly specular surfaces.

PanoLess is designed for mirror-like surfaces: the unity reflection strength of Eq. 8 holds for chrome, polished metal, and glass, but breaks down as surfaces become more diffuse.

To evaluate how the method generalizes beyond this regime, we test on five scenes from the Shiny Blender dataset [30]—*Car*, *Teapot*, *Coffee*, *Helmet*, and *Toaster*—which exhibit a mixture of diffuse and specular materials. Since the original dataset provides dense 360° coverage, which is not the problem we address, we construct **Partial Shiny Blender** by selecting a single-hemisphere

subset of the available camera trajectories (S3), matching the partial-view conditions of our Shiny Partial benchmark.

Qualitative comparisons on Partial Shiny Blender appear in the supplementary. (S3) PanoLess continues to produce structurally consistent environment maps, albeit less coherent than on Shiny Partial: when diffuse albedo is prominent, the Gaussians absorb it into the albedo channel, diverting signal away from the cubemap and yielding a darker, less faithful reconstruction.

Fig. 8 shows an extreme case. The matte coffee surface produces no specular reflections, so the cubemap receives zero gradient from those pixels and the recovered environment map is dim and incomplete. This illustrates the fundamental trade-off: the unattenuated cubemap pathway that drives PanoLess on specular surfaces fails when reflections are absent.

Sensitivity to pose error. Because every cubemap query is indexed by a reflection direction computed from the surface normal, PanoLess is sensitive to camera-pose accuracy, as predicted by our first-order analysis (Eq. 6). A pose-jitter sweep on *Vase* (S5) shows envmap PSNR falling sharply from 18.4 dB to 12.0 dB at only $\sigma=0.5^\circ$ of camera rotation. Accurate poses—e.g. well-converged COLMAP reconstructions as used in our real captures—are therefore a practical prerequisite.

5 Conclusion

We presented PanoLess, a framework that treats highly specular surfaces as wide-angle cameras into the surrounding scene, reconstructing environment illumination from partial views without requiring camera panning or panoramic capture. By combining surface-aligned 2D Gaussian splats with a jointly optimized neural cubemap and an explicit visibility map, PanoLess recovers high-frequency environment maps from a single hemisphere of observations, achieving a nearly 5 dB improvement over the best baseline with mean normal angular error below 5° . Results on both synthetic and real-world captures confirm that the method generalizes across scenes, and the visibility map provides a principled indicator of which environment directions are trustworthy versus extrapolated, enabling informed downstream use in relighting, AR insertion, and scene understanding.

Acknowledgments. This work was supported in part by the Ken Kennedy Institute at Rice University.

References

1. Basri, R., Jacobs, D.W.: Lambertian reflectance and linear subspaces. *IEEE TPAMI* **25**(2), 218–233 (2003)
2. Blinn, J.F., Newell, M.E.: Texture and reflection in computer generated images. *Communications of the ACM* **19**(10), 542–547 (1976)

3. Chen, D., Li, H., Ye, W., Wang, Y., Xie, W., Zhai, S., Wang, N., Liu, H., Bao, H., Zhang, G.: Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics* (2025)
4. Dai, P., Xu, J., Xie, W., Liu, X., Wang, H., Xu, W.: High-quality surface reconstruction using gaussian surfels. In: *SIGGRAPH* (2024)
5. Debevec, P.: Rendering synthetic objects into real scenes: global illumination and hdr photography. In: *SIGGRAPH*. pp. 189–198 (1998)
6. Debevec, P., Hawkins, T., Tchou, C., Duiker, H.P., Sarokin, W., Sagar, M.: Acquiring the reflectance field of a human face. In: *SIGGRAPH*. pp. 145–156 (2000)
7. Gao, J., Gu, C., Lin, Y., Li, Z., Zhu, H., Cao, X., Zhang, L., Yao, Y.: Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing. In: *ECCV* (2024)
8. Gu, C., Wei, X., Zeng, Z., Yao, Y., Zhang, L.: Irgs: Inter-reflective gaussian splatting with 2d gaussian ray tracing. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2025)
9. Hadadan, S., Chen, S., Zwicker, M.: Neural radiosity. *ACM Transactions on Graphics* **40**(6), 1–11 (2021)
10. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: *SIGGRAPH* (2024)
11. Ikeuchi, K.: Determining surface orientations of specular surfaces by using the photometric stereo method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-3**(6), 661–669 (1981)
12. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: *CVPR*. pp. 406–413 (2014)
13. Ji, C., Li, Y., Liao, Y.: Reframe: Reflective surface real-time rendering for mobile devices. In: *European Conference on Computer Vision (ECCV)* (2024)
14. Jiang, Y., Tu, J., Liu, Y., Gao, X., Long, X., Wang, W., Ma, Y.: Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In: *CVPR* (2024)
15. Kautz, J., Sloan, P.P., Snyder, J.: Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics. In: *Eurographics Workshop on Rendering*. pp. 291–296 (2002)
16. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. In: *SIGGRAPH* (2023)
17. Li, Z., Xu, Z., Ramamoorthi, R., Sunkavalli, K., Chandraker, M.: Learning to reconstruct shape and spatially-varying reflectance from a single image. In: *SIGGRAPH Asia* (2018)
18. Liang, R., Chen, H., Li, C., Chen, F., Panneer, S., Vijaykumar, N.: Envidr: Implicit differentiable renderer with neural environment lighting. In: *ICCV*. pp. 79–89 (2023)
19. Liang, Z., Zhang, Q., Feng, Y., Shan, Y., Jia, K.: Gs-ir: 3d gaussian splatting for inverse rendering. In: *CVPR* (2024)
20. Liu, Y., Wang, P., Lin, C., Long, X., Wang, J., Liu, L., Komura, T., Wang, W.: Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. In: *ACM SIGGRAPH 2023 Conference Proceedings* (2023)
21. Meng, J., Li, H., Wu, Y., Gao, Q., Yang, S., Zhang, J., Ma, S.: Mirror-3dgs: Incorporating mirror reflections into 3d gaussian splatting. *arXiv preprint arXiv:2404.01168* (2024)
22. Munkberg, J., et al.: Extracting triangular 3d models, materials, and lighting from images. In: *CVPR* (2022)

23. Oxholm, G., Nishino, K.: Multiview shape and reflectance from natural illumination. In: CVPR (2014)
24. Phongthawee, P., Chinchuthakun, W., Sinsunthithet, N., Jampani, V., Raj, A., Khungurn, P., Suwajanakorn, S.: Diffusionlight: Light probes for free by painting a chrome ball. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
25. Ramamoorthi, R., Hanrahan, P.: A signal-processing framework for inverse rendering. In: SIGGRAPH. pp. 117–128 (2001)
26. Sloan, P.P., Kautz, J., Snyder, J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In: SIGGRAPH. pp. 527–536 (2002)
27. Somanath, G., Kurz, D.: Hdr environment map estimation for real-time augmented reality. In: CVPR (2021)
28. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer (2010)
29. Turkulainen, M., Ren, X., Melekhov, I., Seiskari, O., Rahtu, E., Kannala, J.: Dn-splatter: Depth and normal priors for gaussian splatting and meshing. arXiv preprint arXiv:2403.17822 (2024)
30. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-nerf: Structured view-dependent appearance for neural radiance fields. In: CVPR. pp. 5481–5490 (2022)
31. Xiang, H., Li, X., Cheng, K., Lai, X., Zhang, W., Liao, Z., Zeng, L., Liu, X.: Gaussianroom: Improving 3d gaussian splatting with sdf guidance and monocular cues for indoor scene reconstruction. In: IEEE International Conference on Robotics and Automation (ICRA) (2025)
32. Xie, T., Chen, X., Xu, Z., Xie, Y., Jin, Y., Shen, Y., Peng, S., Bao, H., Zhou, X.: Envgs: Modeling view-dependent appearance with environment gaussian. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
33. Yan, Y., et al.: Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In: ECCV (2024)
34. Yang, Z., Gao, X., Sun, Y., Huang, Y., Lyu, X., Zhou, W., Jiao, S., Qi, X., Jin, X.: Spec-gaussian: Anisotropic view-dependent appearance for 3d gaussian splatting. In: NeurIPS (2024)
35. Yao, Y., Zeng, Z., Gu, C., Zhu, X., Zhang, L.: Reflective gaussian splatting. In: International Conference on Learning Representations (ICLR) (2025)
36. Ye, K., Hou, Q., Zhou, K.: 3d gaussian splatting with deferred reflection. In: SIGGRAPH (2024)
37. Yu, Z., Sattler, T., Geiger, A.: Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. ACM Transactions on Graphics (TOG) (2024)
38. Zhang, W., Tang, J., Zhang, W., Fang, Y., Liu, Y.S., Han, Z.: Materialrefgs: Reflective gaussian splatting with multi-view consistent material inference. NeurIPS (2025)
39. Zhang, Y., Chen, A., Wan, Y., Song, Z., Yu, J., Luo, Y., Yang, W.: Ref-gs: Directional factorization for 2d gaussian splatting. In: CVPR (2025)