

Beyond Spherical Harmonics: Rethinking Appearance Models for Radiance Reconstruction

Ewa Miazga^{1,2}  Jorge Condor²  Piotr Didyk² 

¹École Polytechnique Fédérale de Lausanne, Switzerland

²Università della Svizzera Italiana, Lugano, Switzerland

ewa.miazga@epfl.ch, {ewa.miazga, jorge.condor, piotr.didyk}@usi.ch

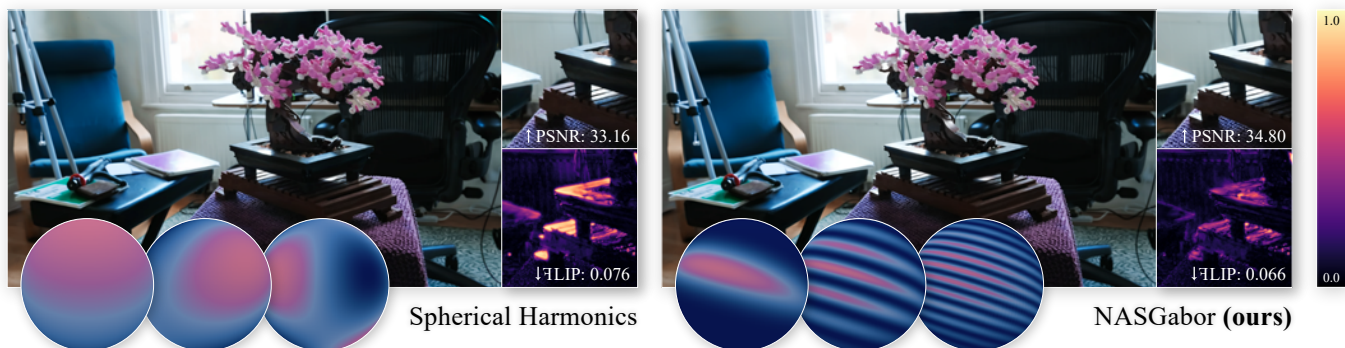


Figure 1: We introduce a new spherical function, the Normalized Anisotropic Spherical Gabor (NASGabor), an anisotropic, multi-modal kernel with a closed-form integral expression. The proposed representation is both compact and highly expressive, and is particularly well-suited for modeling view-dependent effects in novel view synthesis, outperforming commonly used approaches such as Spherical Harmonics.

Abstract

View-dependent appearance modeling remains a challenging problem in novel-view synthesis and reconstruction. Accurately representing complex angular effects often requires substantial memory and computational resources. For new learning-based methods, a common approach is to rely on Spherical Harmonics (SH). However, capturing high-frequency phenomena such as specular reflections demands high-order expansions, which increase memory usage and computational cost. Consequently, most methods employ low-order SH, which limits the ability to model complex view-dependent effects, resulting in overly smooth or diffuse representations. To address these limitations, we systematically evaluate a wide range of spherical functions in the context of scene reconstruction. Some of them are introduced to graphics and computer vision for the first time in this paper. Based on the insights from the experiment, we develop a novel spherical formulation, the Normalized Anisotropic Spherical Gabor function that enables efficient modeling and learning of high-frequency appearance effects while maintaining compact representation. Compared to existing approaches, our function achieves higher-quality reconstruction of view-dependent phenomena such as glints, while being up to five times more memory-efficient and more efficient to evaluate. We validate its performance in radiance-field reconstruction tasks.

CCS Concepts

• *Computing methodologies* → *Reflectance modeling; Appearance and texture representations; Point-based models;*

1. Introduction

Despite their success, existing methods exhibit fundamental limitations in modeling view-dependent appearance. Most approaches

rely on low-order spherical harmonics basis (Spherical Harmonics (SH)) to encode directional radiance, which inherently restricts their expressive power. SH are frequency-limited, restricted to low

frequency angular appearance modeling; while they are an efficient basis for representing smooth, diffuse signals, they fail to capture high-frequency effects, leading to blurred or inaccurate highlight reconstruction. This limitation stems from both the band-limited nature of SH and the joint modelling of geometry and appearance.

At the same time, the computer graphics literature has extensively explored different spherical functions and bases to model view-dependent appearance in many different applications, ranging from path guiding [HIT*24, XSD*13], real-time global illumination approximations [HDHN16, KHDN22], approximating and sampling environment map lighting [RPFGLM23, XSD*13], point-based global illumination [VADWG15, SKS02, GKMD06, GKD07, HSRG07, LWDB10, RBS*11, YZXW12, IFDN12], shadow and visibility approximations [XSD*13, NRH03, Ram06, RWS*06] and BRDFs [WZT*08, War92, AS01, CT82, TS67, NDM05, PSCS*12] among many others. However, they have seen limited adoption in scene reconstruction, despite the clear limitations of SH.

In this work, we revisit the problem of view-dependent appearance modeling in novel view synthesis. We systematically investigate a wide range of spherical distributions, many of them new to the graphics literature, and analyze their suitability in the task. We further develop closed-form integrals and analytic derivatives for a number of them. Through this analysis, we identify the key properties that govern their effectiveness in this task: multi-modality, closed-form integration and anisotropy. Next, guided by these properties, we introduce a novel spherical function, the Normalized Anisotropic Spherical Gabor kernel, that fulfills them, achieving state-of-the-art results in scene reconstruction in terms of quality, at a fraction of the memory cost of SH, and increased rendering speed. It enables accurate reconstruction of complex view-dependent effects such as specular highlights and is easy to integrate in any 3DGS-based pipeline.

2. Related Work

2.1. Novel View Synthesis with Radiance Fields

Light fields have been effectively used for a long time in the context of image-based rendering, abstracting light transport modeling and enabling interpolation between input views [LH96]. They have seen growing interest in recent years, primarily thanks to the introduction of neural networks [HPP*18, SZW19]. In particular, Neural Radiance Fields [MST*21] and derivative works [BMT*21, MBRS*21, MESK22] popularized the field and enabled high-quality reconstruction, at a reasonable computational cost. Neural methods, while flexible and capable of modeling complex signals, do not scale well. Additionally, their lack of an explicit geometrical structure makes them challenging to edit and integrate into dynamic systems.

More recently, point-based methods, either rasterized or ray-traced, have displaced implicit fields in radiance-field rendering [KKLD23, WYF*24, CSB*25, MLMP*24]. Their explicit representation enables better scalability and generally faster rendering, as well as easier editability, among many other benefits. However, their joint modeling of geometry and appearance makes them memory-intensive and difficult to scale for high-frequency

detail [KMK*24], both spatially and in the view direction domain. Several hybrid approaches combining implicit fields, voxel grids, and Lagrangian particles have also been proposed [RPLG21, LSS*21, STC*22, CMLND*26, KLR*22, ZNM*26, YLT*21]. Despite all their limitations, they seek a balance between run-time speed, compression, quality, and scalability. Additionally, primitive-based methods are particularly interesting for relighting, second-order effects such as shadowing and reflections, and physically-based rendering, when combined with environment maps and intrinsic decomposition networks [LHG*25, DLDN25, WSCH26, CSB*25]

2.2. Appearance Modeling in Radiance Fields

There is a wide range of appearance modeling strategies for radiance fields. Implicit neural fields typically rely on their MLPs to encode view-dependent radiance [MST*21]. Voxel-based representations store learnable features on a grid and often rely on low-degree spherical harmonics (SH) [STC*22], but can also use implicit features and neural shading [MHK*26, CFHT23, YHR*23]. Similarly, most point-based approaches, such as 3DGS [KKLD23], use low-degree SH to approximate directional radiance. While this enables fast, relatively compact rendering, the low degree of SH limits modeling power, making the reproduction of high-frequency view-dependent effects, such as specular highlights and reflections, particularly difficult.

To address these limitations, more expressive spherical functions have been recently proposed. Beta-Splatting [LSC*25] utilizes Spherical Betas (SB) [Tre96], a function similar to isotropic spherical Gaussians, but with controllable sharpness. Concurrent work [DSRV*26] has proposed using spherical Voronoi [NLC02] (SV) models, which are learnable partitions of the sphere with tunable smoothness. Some approaches combine explicit models, like anisotropic spherical Gaussian mixtures, with universal function approximators (MLPs) for maximum flexibility [YGS*24]. Alternatively, view-dependence can be introduced into the density field itself by enhancing opacity weights with SH basis [ZWZW25]. More recent concurrent work on Neural Harmonic Textures [CMLND*26] proposes using neural deferred shading and small explicit local fields to model view-dependent radiance and detach appearance from geometry. While these works demonstrate substantial quality improvements using more expressive appearance models, the methods still exhibit clear trade-offs between compactness, expressivity, computational efficiency, and interpretability. In our work, we explore a wide variety of parametric spherical functions to understand the techniques and properties that make an ideal appearance model in the context of radiance fields.

2.3. Parametric Spherical Functions as Appearance Models

In computer graphics, view-dependent appearance and reflectance have long been modeled using parametric spherical functions. Classical reflectance models, such as the Phong or Ward models [Pho98, War92], use different spherical functions to model directional, isotropic, or anisotropic light scattering. Microfacet BRDF models [CT82, WMLT07] rely on distributions of surface normals to characterize reflectance properties. Similar to earlier methods, these normal distribution functions often take the

form of spherical functions such as generalized cosine lobes or Gaussian-like kernels defined on the sphere.

Beyond classical reflectance models, Spherical Gaussians (SG) [Fis53] provide a smooth, differentiable basis for representing directional signals such as illumination, BRDF lobes, and view-dependent color in both traditional and neural rendering pipelines [VADWG15, SKS02, GKMD06, GKD07, HSRG07, LWDB10, RBS*11, YZXW12, IFDN12]. However, isotropic SG mixtures cannot capture strongly anisotropic effects efficiently, and are not very expressive [RPFGLM23]. Anisotropic Spherical Gaussians (ASGs) [XSD*13] and recently proposed Normalized ASGs [HIT*24] extend SGs and have been used in applications such as path guiding [MGN17]. Similarly, linearly transformed cosines (LTC) [HDHN16] and their anisotropic extensions [KHDN22] have been employed for fast lighting approximations. More recently, log-space Spherical Harmonics basis functions have also been employed in the context of environment map regression and approximate global illumination [SSIN25].

Beyond the graphics literature, a wide range of fields, from geology to physics and statistics, employ families of functions defined on the surface of the sphere for various purposes, such as gravity [MV25] and weather system [WHM13] simulations. We explore some of these functions [KM20, Fis53, Yua20, MG20], introducing them to the graphics literature for the first time and assessing their potential for radiance-field rendering and beyond.

3. Background

This section provides background on primitive-based radiance fields, which are the main focus of this paper, as well as on spherical harmonics, the most common choice for modeling appearance in such representations.

3.1. Primitive-based Radiance Fields

The goal of novel view synthesis is to reconstruct a 3D scene from a set of input images and generate photorealistic images from previously unseen viewpoints.

Primitive-based methods [LSS*21] model a scene as a collection of explicit primitives rather than a continuous implicit field [MST*21, BMV*23]. In particular, point-based approaches like 3D Gaussian Splatting [KKLD23] represent the scene using a set of primitives $\mathcal{P} = \{p_i\}$, where each primitive encodes both geometry and appearance. A common choice are anisotropic 3D Gaussians, which provide a compact and differentiable representation of local scene structure. Each primitive p_i is parameterized by a mean $\mu_i \in \mathbb{R}^3$ and a covariance matrix $\Sigma_i \in \mathbb{R}^{3 \times 3}$, defining its spatial extent and pose, and employing a Gaussian density function as its kernel:

$$\rho_i(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^\top \Sigma_i^{-1}(\mathbf{x} - \mu_i)\right). \quad (1)$$

To ensure positive semi-definiteness, the covariance is typically parameterized as $\Sigma_i = \mathbf{R}_i \mathbf{S}_i \mathbf{S}_i^\top \mathbf{R}_i^\top$, where \mathbf{R}_i is a rotation matrix and \mathbf{S}_i is a diagonal scaling matrix, following SVD decompositions. In addition to geometry, each primitive carries an opacity σ_i and

a view-dependent radiance function $\mathbf{c}_i(\mathbf{d})$, often represented using SH to capture directional appearance effects.

Given a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, rendering is performed by alpha-blending contributions from primitives along the ray. The resulting color is computed using a discrete volume rendering formulation:

$$C(\mathbf{r}) = \sum_{i \in \mathcal{P}_r} T_i \alpha_i \mathbf{c}_i(\mathbf{d}), \quad (2)$$

where \mathcal{P}_r denotes the set of primitives contributing to the ray. The opacity of each primitive is defined as

$$\alpha_i = \sigma_i \rho_i(\mathbf{r}(t_i)), \quad (3)$$

and the transmittance is given by

$$T_i = \prod_{j < i} (1 - \alpha_j). \quad (4)$$

This formulation can be interpreted as a discretized approximation of the classical volume rendering equation, where continuous density fields are replaced by a finite set of localized primitives. In contrast to neural implicit representations, primitive-based methods avoid costly per-sample network evaluations and instead rely on explicit representations that can be efficiently rasterized and projected in an approximate manner onto the image plane.

3.2. Spherical Harmonics

Spherical Harmonics (SH) are a classical tool for representing magnitudes defined over the sphere, and are widely used to model view-dependent appearance in primitive-based radiance fields. They express a directional signal as a weighted sum of fixed, orthogonal basis functions:

$$f(\mathbf{d}; c) = \sum_{l=0}^L \sum_{m=-l}^l c_{lm} Y_l^m(\mathbf{d}), \quad (5)$$

where $Y_l^m(\mathbf{d})$ are the spherical harmonic basis functions and c_{lm} are the corresponding coefficients.

A key advantage of SH is that appearance modeling reduces to learning only the coefficients c_{lm} , while the basis functions remain fixed, making the representation computationally efficient, easy to optimize, and fast to evaluate at rendering time. However, SH representations are inherently frequency-limited by the number of basis functions used. Low-degree SH provide a compact approximation of smooth, low-frequency signals, making them particularly well-suited for diffuse color and softly varying illumination. Capturing high-frequency, view-dependent effects, such as sharp specular highlights or complex directional reflectance, requires increasing the degree L , leading to rapid growth in the number of coefficients and basis evaluations and, in turn, increasing computational cost and memory requirements. Consequently, practical systems sacrifice the quality of view-dependent appearance by typically restricting SH to low orders.

4. Analyzing Spherical Functions for Radiance Fields

In this section, we describe our study of different spherical functions in the context of appearance reproduction for radiance fields.

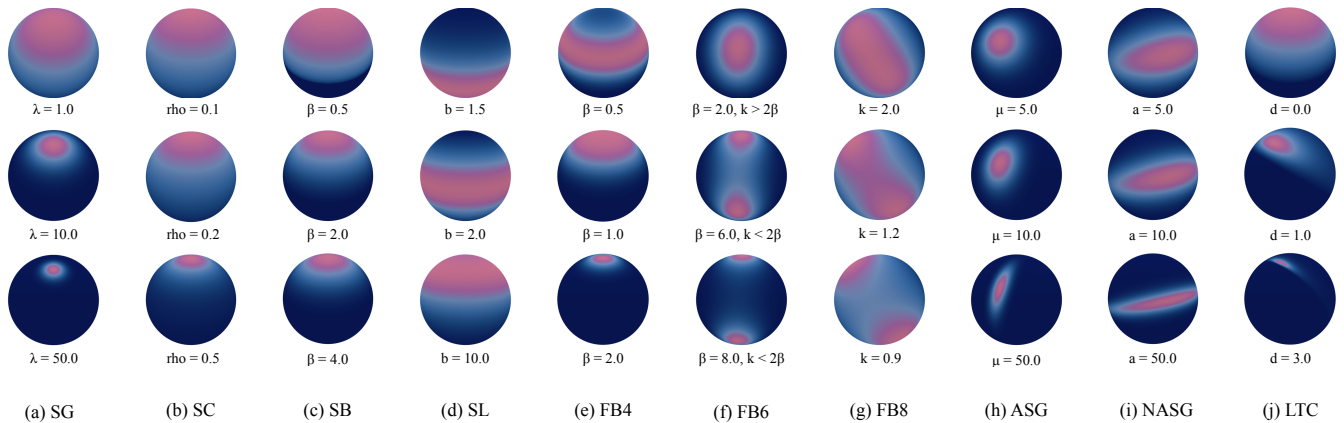


Figure 2: Visualization of the distributions used in the evaluation, showing the effect of varying parameters. Each row varies a single parameter while keeping the others fixed, highlighting how the distributions behave. The peak of each distribution can be positioned anywhere on the sphere. Isotropic functions: (a) Spherical Gaussian (SG) [Fis53], (b) Spherical Cauchy (SC) [KM20], (c) Spherical Beta (SB) [Tre96]. Ring-like and bimodal functions: (d) Spherical Logistic (SL) [MG20], (e) Fisher Bingham 4-parameter (FB4) [BM78], (f) Fisher Bingham 6-parameter (FB6) [Yua20], (g) Fisher Bingham 8-parameter (FB8) [Yua20]. Anisotropic functions: (h) Anisotropic Spherical Gaussian (ASG) [XSD*13] (i) Normalized Anisotropic Spherical Gaussian (NASG) [HIT*24], (j) Linearly Transformed Cosine [HDHN16].

We describe our choice of models, the experimental setup, and discuss our findings.

4.1. Parametric Spherical Functions

To investigate new models of view-dependent appearance, we explored a wide range of spherical functions, where some are new to the graphics literature. We generally chose compact parametric functions, for which closed-form integration is possible, and that are relatively inexpensive to evaluate, balancing expressive power and computational efficiency. We group them according to their functional properties and the types of effects they can represent.

Isotropic Functions. In Figure 2: (a-c) are symmetric around a peak direction and unimodal. They are the most compact and efficient to evaluate.

Anisotropic Functions. In Figure 2: (h-j). They introduce shape anisotropy and arbitrary rotations, enabling higher modeling power at minimal increase in complexity.

Ring-like and Bimodal Functions. In Figure 2: (d-g). Ring-like functions form high-density regions along a small circle. Bimodal functions have two coupled lobes over an axis of symmetry, effectively allowing a single function to capture multiple directional features. In the case of the 8-parameter Fisher-Bingham distribution, the lobes are also anisotropic. This multi-moded characteristic enhances the expressive power of the model, enabling it to approximate structured or repeated reflection patterns using fewer components.

All evaluated distributions grouped by family are shown in Figure 2. For each function, we provide the probability density function (PDF) and, if available, closed-form integral expressions in

Scene	d_{3nn}	SH	Default		+ Auto LR Init	
			NASG	Δ	NASG	Δ
Indoor	0.303	31.62	32.15	+0.53	32.08	+0.46
Outdoor	0.621	24.71	24.21	-0.50	24.51	-0.20

Table 1: Effect of automatic learning rate initialization on reconstruction quality (PSNR) for NASG [HIT*24]. Results are reported for indoor and outdoor scenes along with the average camera spacing d_{3nn} .

Supplementary Material A. Additionally, we derived the closed-form integral for the Spherical Beta function [Tre96] (Supplementary Material D).

4.2. Implementation Details

We build upon the `gsplat` [YLK*25] framework, in particular using Beta Splatting [LSC*25]’s parametric opacity kernels. We use the same appearance model for all spherical functions:

$$f_C(\mathbf{d}) = \mathbf{c}_0 + \sum_{l=1}^L \mathbf{w}_l G_l(\mathbf{d}), \quad (6)$$

where \mathbf{c}_0 is a diffuse RGB color, G_l the spherical function, L the number of lobes and $\mathbf{w}_l \in \mathbb{R}^3$ a learned weight. In all of our tests, we have a fixed budget of up to 1M primitives per scene and we train for a fixed number of iterations, using the same hyperparameters.

Generally, more expressive appearance models can overfit under sparse observations, which becomes visible when interpolating between views [DSRV*26]. This can be mitigated by choosing adequate learning rates. Instead of following recent work and manually adapting learning rates per dataset [DSRV*26, LSC*25], we

develop a simple heuristic based on the average distance between cameras in world-space and their k -nearest neighbors d_{knn} , making it broadly applicable to any dataset. The underlying idea is to choose smaller learning rates for the spherical parameters as observations become sparser, dampening the higher modeling power. We empirically determined base learning rates for the spherical parameters or features $\eta_{f,b}$ and kernel opacities $\eta_{\sigma,b}$. We define a reference distance d_{ref} as the average k -nearest neighbor distance between camera centers in the `bonsai` scene. We then define

$$\eta_f = \eta_{f,b} \left(\frac{d_{\text{ref}}}{d_{\text{knn}}} \right)^\beta \quad (7)$$

$$\eta_\sigma = \eta_{\sigma,b} \left(\frac{d_{\text{knn}}}{d_{\text{ref}}} \right)^\gamma \quad (8)$$

with constants $\beta = 2$, $\gamma = 0.6$ and using $k = 3$ nearest neighbours. We show the effect of the heuristic in Table 1.

4.3. Datasets

We evaluate all the proposed view-dependent functions on the *Mip-NeRF360* [BMV*22] dataset, which provides a diverse collection of high-resolution indoor and outdoor scenes. A key difficulty in outdoor scenes arises from the aforementioned sparsity of training views, which we partially compensate through our learning rate scaling heuristic.

4.4. Results and Discussion

The main results of this evaluation are presented in Figure 3.

Interpolation Quality vs Expressiveness. Our first conclusion follows our prior intuition: increasing the expressiveness of the functions, e.g., through multi-lobe formulations or highly flexible functions, does not lead to consistent improvements and can even degrade generalization due to poor interpolation between views in sparsely captured datasets (outdoor scenes in MipNeRF360 [BMV*22]).

In contrast, SH demonstrates strong robustness in this setting. Although they are inherently limited in representing high-frequency effects such as sharp highlights, their low-frequency nature results in smooth angular interpolation. This leads to a stable and consistent appearance across unseen viewpoints, making SH particularly effective under sparse view sampling. Quantitatively, though, the performance gap between SH and more expressive spherical functions remains small (around 0.2dB of PSNR), indicating that all methods achieve comparable reconstruction quality in this setup.

Indoor scenes present a complementary scenario, where camera views are denser. In this case, the training data provides sufficient angular coverage to reconstruct complex view-dependent effects, enabling more expressive spherical functions to realize their full potential. We observe that increasing the complexity of the appearance model consistently improves reconstruction quality. In particular, anisotropic functions provide substantial gains, thanks to their increased flexibility.

Quality vs Memory Compactness. Among the evaluated approaches, the NASG [HIT*24] function achieves the best performance, demonstrating the importance of anisotropy. We note that for this experiment, we used a slightly different parameterization to improve compactness and optimization stability than that provided in the original paper [HIT*24]; we include more details in Supplementary Material C. NASG [HIT*24] is among the most parameter-intensive functions we evaluated, which shows the trade-off between compactness and quality, largely shown across all results.

We observe that learnable spherical functions are more efficient than SH. In particular, a single lobe of spherical functions combined with a diffuse component yields comparable or superior performance in many instances, while requiring approximately $5\times$ fewer parameters than commonly used 3rd-degree SH.

Multi-modal and Multiple-lobed Distributions. Multi-lobe formulations further enhance performance by allowing the representation of more complex angular distributions, but the impact of adding extra lobes has diminishing returns. Multiple lobes can collapse to a single position due to the complex optimization landscape. In Figure 4, we explore adding a regularization based on the Spherical Wasserstein Distance (SWD) [BBC*23] to discourage lobe similarity. While this made the appearance distribution more diverse, it did not improve final performance. In many instances, it led to the collapse of one or more lobes, indicating that it may be an interesting avenue to explore for compression.

Interestingly, we note that normalized multi-modal distributions like FB6 [Yua20] outperform similarly shaped normalized unimodal distributions like Spherical Gaussians (SG) [Fis53]. We hypothesize it is easier for the optimization to leverage multiple lobes when they are coupled through a joint parameterization; we will take this insight to the extreme in our proposed novel function in Section 5.

Effect of Normalization. Although mostly overlooked in appearance modeling for novel-view synthesis literature, normalization plays a fundamental role in achieving a stable optimization, by improving the conditioning of the underlying learning problem. In machine learning, it has been shown that transforming inputs toward zero mean and unit variance leads to a better-conditioned optimization, thereby facilitating faster and more stable convergence [LBOM12, Sch12]. In deep neural networks, normalization techniques such as Batch Normalization help maintain well-behaved activation distributions across layers, preventing pathological curvature and unstable gradients during training [IS15, Glo10, HZRS15].

We observe an analogous effect in our context. Normalizing spherical functions stabilizes optimization by improving gradient behavior. This is particularly important for functions defined using exponential terms, where outputs can grow rapidly and lead to numerical instability. We study the effect of normalization in Figure 5. In multi-lobe representations, enforcing normalization reduces degeneracy and improves numerical conditioning, leading to more stable convergence. This effect is particularly pronounced in some cases such as Spherical Cauchy (SC) [KM20] and 6-parameter Fisher–Bingham functions (FB6) [Yua20].

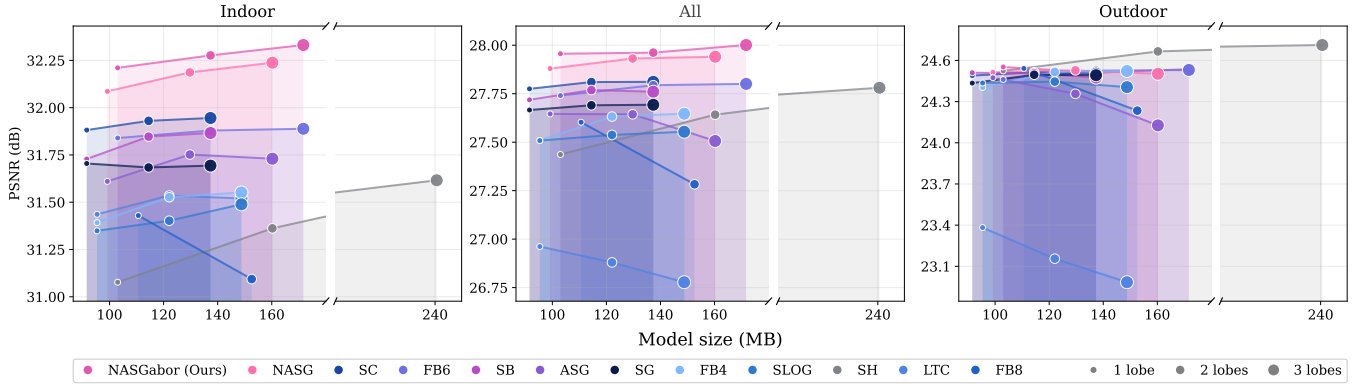


Figure 3: Comparison between reconstruction quality (PSNR) and model memory (MB) for different spherical functions showed in Figure 2 with varying number of lobes, grouped by scene type. Graphs showing $FLIP$ [ANAM*20], $FovVDP$ [MDC*21] and MSE metric can be found in Supplementary Material B.

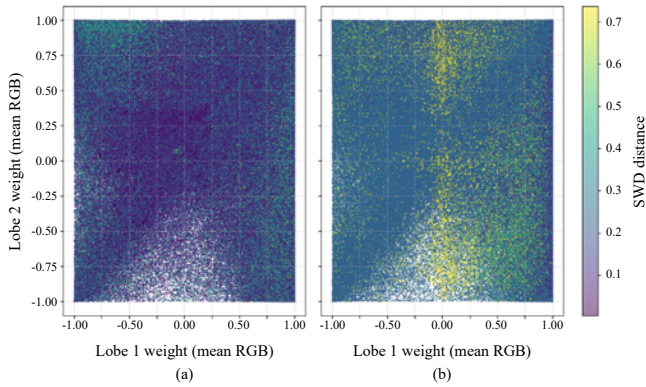


Figure 4: The x - and y -axes show the corresponding lobe weights, representing each lobe’s contribution, where a value of 0.0 indicates no contribution (i.e., the lobe is effectively deactivated). (a) shows the learned weights after optimization without SWD regularization, while (b) shows the result with SWD regularization. Although the regularization encourages diversity among lobes, it also suppresses the contribution of one of the lobes during optimization, leading to degraded performance.

However, the cost and accuracy of normalization vary significantly across function families. For the Fisher-Bingham family, computing normalization constants is computationally expensive, and in the case of Fb6 and FB8 [Yua20] only approximate normalization is available. Due to the resulting training overhead, we report convergence results for the unnormalized variant of FB8 [Yua20] in Figure 3. Generally, only when relatively simple closed-form integral expressions are available, we could consider them practical for our optimization problem.

Key findings From the above analysis, we distill three key findings:

1. Anisotropic functions are notably superior to isotropic kernels for the task.

2. Multiple coupled lobes can be leveraged better for quality than independent lobes.
3. Closed-form integral expressions need to be available and ideally be relatively efficient to compute.

5. A Novel Anisotropic Appearance Model

Inspired by our findings in the previous section, we propose the Normalized Anisotropic Spherical Gabor (NASGabor) function as a new appearance model.

5.1. Normalized Anisotropic Spherical Gabor

The new function is a composition of a NASG [HIT*24] spherical envelope with a harmonic carrier (cosine), shifted to ensure positive values across the entire range. The formulation is as follows:

$$G(\mathbf{d}; [\mathbf{x}, \mathbf{y}, \mathbf{z}], \lambda, a, k) = \begin{cases} \frac{1 + \cos(k \mathbf{d} \cdot \mathbf{x})}{2} \exp\{2\lambda \kappa^{1+\tau} - 2\lambda\} \kappa^\tau, & \mathbf{d} \neq \pm \mathbf{z}, \\ 1, & \mathbf{d} = \mathbf{z}, \\ 0, & \mathbf{d} = -\mathbf{z}, \end{cases} \quad (9)$$

$$\kappa = \frac{\mathbf{d} \cdot \mathbf{z} + 1}{2}, \quad \tau = \frac{a(\mathbf{d} \cdot \mathbf{x})^2}{1 - (\mathbf{d} \cdot \mathbf{z})^2}.$$

We also derive its integral in closed form, with the following expression:

$$\int_0^{2\pi} \int_0^\pi G(\mathbf{d}; [\mathbf{x}, \mathbf{y}, \mathbf{z}], \lambda, a, k) \sin \phi \, d\phi \, d\theta = \frac{\pi(1 - e^{-2\lambda})}{\lambda \sqrt{1+a}} (1 + \Psi) \quad (10)$$

$$\Psi = \frac{2\Lambda e^{-\Lambda} \operatorname{sinhc}(\sqrt{\Lambda^2 - k^2})}{1 - e^{-2\Lambda}}, \quad \Lambda = \lambda(1+a)$$

with sinhc being the hyperbolic sinc function. The function

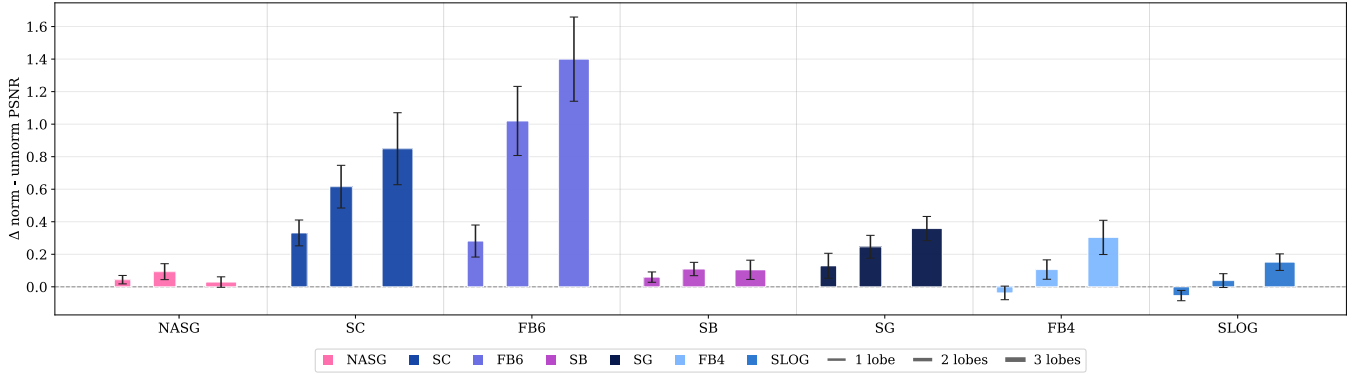


Figure 5: Comparison of normalized and unnormalized distributions across varying numbers of lobes. Colored bars show the mean PSNR difference, with whiskers indicating the standard error of the mean across scenes. Learning normalized distributions generally improves final quality across the board.

is parameterized by an orthonormal frame $[x, y, z]$ and a set of shape-adjusting parameters. Broadly, λ controls the spread of the function over the sphere, a is an anisotropy/sharpness parameter, and k controls the frequency of the carrier, effectively changing the number of ripples. We visualize the kernel in Figure 1. The addition of a harmonic function creates a controllable number of ripples; a strategy recently pursued to augment the modeling power of base Gaussian opacity kernels, in the contexts of radiance field [ZHD*25, WZZ24] and physically based rendering [CHYD26].

Similar to our experiment in Section 4, we compose lobes of NASGabor with diffuse color, which generally helps in encouraging the spherical distributions to focus on angular detail, resulting in the same appearance model presented in Equation 6.

We run NASGabor in our former experiment and include it in Figure 3, outperforming all previous functions in MipNerf360 [BMV*22].

Since the analytic integral of NASGabor is relatively more expensive than un-modulated NASG [HIT*24], we tested optimizing with an approximation of it as well, one that disregards the carrier function. This produced similar results in terms of quality. For low frequencies, the approximation is reasonably accurate, and allows us to train significantly faster. We explore this choice and include results for both in Table 2. All other results in Section 6 are done with the approximated integral.

Table 2: Efficiency comparison between 2-lobed NASGabor using the exact normalization, and the approximate one (which disregards the effect of the carrier function), on MipNerf360 [BMV*22]. The approximation is notably accurate at low frequencies, becoming exact at $k = 0$, and enables us to train faster.

Method	PSNR \uparrow	Train Time \downarrow	textbfFPS \uparrow
Approximate	28.46	14m42s	143
Exact	28.47	20m20s	145

5.2. Implementation Details

To maximize optimization and rendering efficiency, we compute analytical derivatives for the NASGabor function (Supplementary Material F). We also include analytical derivatives for the unmodulated NASG [HIT*24] function (Supplementary Material E). We implement the appearance model within the `gsplat` framework [YLK*25] through a pair of extra CUDA kernels, handling backward and forward passes. As with 3DGS [KKLD23] and all derivative works, we compute the response of each primitive along a central camera ray for each frame, then alpha-blend them in sorted order. This is a convenient approximation, reducing the load drastically from one evaluation per intersection to one per primitive, per frame. Additionally, we found that employing a learning rate scheduler further stabilizes the optimization and improves final results, similarly to how Gaussian means are already scheduled in 3DGS. We include further details on parametrization and the scheduler in the Supplementary Material C.

6. Results

We evaluate the performance of our spherical function against recent work on appearance models for primitive-based radiance fields [LSC*25, DSRV*26, IKK*24]. To ensure fairness, all models are implemented in `gsplat` [YLK*25], and tested across a wide range of classical benchmarks: MipNerf360 [BMV*22], Tanks and Temples [KPZK17], and Deep Blending [HPP*18]. We show quantitative results in Table 3, qualitative results in Figure 6, and runtime performance measurements for both training and forward rendering in Table 5. All scenes use the same hyperparameters and training length, except learning rates, which are adjusted either to the values suggested by their respective works (manually tuned) or in our case, using our heuristic (no hand-tuning). We additionally compare against other related works that do not share the same `gsplat` backbone in Table 4.

In the particular case of Spherical Voronoi (SV) [DSRV*26], due to the lack of analytic backward derivatives, two renderers were released by the authors: a `gsplat`-based [YLK*25] renderer for fast evaluation (with a CUDA forward kernel) and a



Figure 6: Qualitative comparison of different appearance models. Our method achieves comparable or superior performance in appearance reconstruction. The insets highlight view-dependent effects that are accurately captured by our appearance model.

Table 3: Quantitative comparison of explicit novel-view synthesis methods conducted on Mip-NeRF360 [BMV*22], DeepBlending [HPP*18], and Tanks&Temples [KPZK17] datasets. *Params* indicates parameters per primitive reserved for appearance

Method	Params	Mip-NeRF360			DeepBlending			Tanks&Temples		
		PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
2DGS [HYC*24] + SH	48	27.22	0.804	0.275	29.56	0.904	0.325	22.85	0.827	0.244
3DGS-MCMC [IKK*24] + SH	48	27.99	0.830	0.229	29.49	0.912	0.306	24.46	0.866	0.174
3DGUT-MCMC [WMEM*25] + SH	48	27.82	0.826	0.233	29.87	0.913	0.309	24.20	0.861	0.180
Beta Splatting [LSC*25] + SH	48	28.00	0.830	0.226	29.79	0.911	0.294	24.34	0.868	0.173
Beta Splatting [LSC*25] + SB	15	28.09	0.830	0.227	29.24	0.908	0.301	24.64	0.870	0.173
Spherical Voronoi [DSRV*26]	12	28.19	0.830	0.230	29.56	0.907	0.301	24.64	0.869	0.197
Spherical Voronoi [DSRV*26]	48	28.46	0.832	0.226	29.90	0.912	0.301	24.67	0.870	0.172
NASGabor – 1 lobe	12	28.40	0.829	0.228	29.91	0.911	0.290	24.64	0.867	0.173
NASGabor – 2 lobes	21	28.46	0.830	0.227	30.24	0.915	0.287	24.68	0.869	0.173
NASGabor – 4 lobes	39	28.46	0.830	0.227	30.37	0.914	0.286	24.79	0.869	0.172

Torch-based renderer used for optimization, combined with 3DGS-MCMC’s [IKK*24] rasterizer. Notably, these two renderers can produce up to a 2 dB difference in PSNR on the same scene checkpoint. For consistency, we report evaluation results and measurements using the Torch-based renderer, which produces better results.

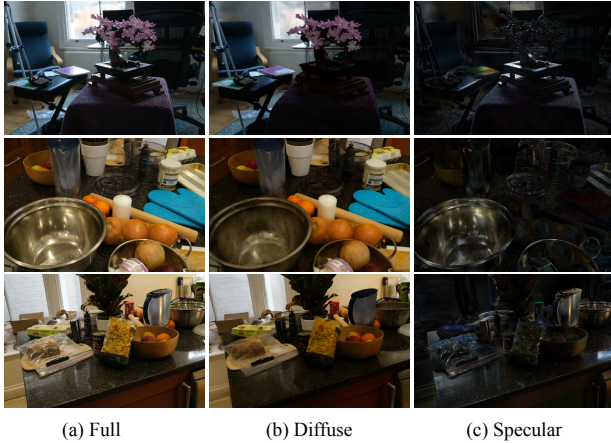
Overall, NASGabor matches or outperforms previous works in terms of quality (Tables 3, 4), at a fraction of the memory cost, enabling it to train and render at faster speeds (Table 5). In Figure 8, we show FLIP [ANAM*20] maps, showcasing how our method

enables more accurate depiction of sheens, highlights, and generally high frequency effects, compared to spherical harmonics. We further include aggregated results for individual scenes and variable numbers of lobes in the Supplementary Material B. All of the experiments were conducted on NVIDIA Grace Hopper (GH200) GPUs.

Disentangling Color Albedo from Specular Effects An interesting byproduct of mixing diffuse color with spherical functions as an appearance model is its capacity to naturally disentangle diffuse albedo from other view-dependent effects related to material

Table 4: Quantitative comparison of novel-view synthesis 3DGS rasterizer-backed methods on Mip-NeRF360 [BMV*22] (7 scenes setting, excluding flowers and treehill, following their reported setup), DeepBlending [HPP*18], and Tanks&Temples [KPZK17]. Storage and FPS are reported only for the Mip-NeRF360 setting.

Method	Mip-NeRF360					DeepBlending			Tanks&Temples		
	Storage	FPS	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Spec-Gaussians [YGS*24]	878 MB	71.6	29.00	0.871	0.207	29.37	0.901	0.314	23.65	0.851	0.202
Glossy-Gaussian [ZWZ25]	888 MB	146.0	28.64	0.858	0.232	28.53	0.890	0.324	23.43	0.846	0.211
NASGabor - 1 lobe	320 MB	148.0	30.14	0.884	0.202	29.91	0.911	0.290	24.64	0.867	0.173



(a) Full (b) Diffuse (c) Specular

Figure 7: (a) Full color reconstruction, decomposed into (b) diffuse and (c) specular components, is naturally supported by our representation, where diffuse appearance is explicitly modeled, and view-dependent effects are captured by the NASGabor functions.

properties or lighting. We showcase it in Figure 7. This basic and free intrinsic decomposition could potentially be used for simple relighting, material estimation or reflections removal.

7. Limitations and Future Work

Increasing the expressivity of primitives in radiance field reconstructions can lead to increased overfitting to the training data, resulting in poor generalization to unseen viewpoints, particularly when the training data is sparse. Data-driven approaches that enable supervision in unseen views [HCD25] could help alleviate this limitation. In this work, we considered working directly with tone-mapped RGB images. Studying how different appearance modeling strategies perform for high-dynamic-range content is an interesting direction for future work [MHMB*21, WWK*24]. Furthermore, the Normalized Anisotropic Spherical Gabor function can be beneficial for a variety of problems involving spherical function representations, such as path-guiding, approximate global illumination, and environment map compression and sampling.

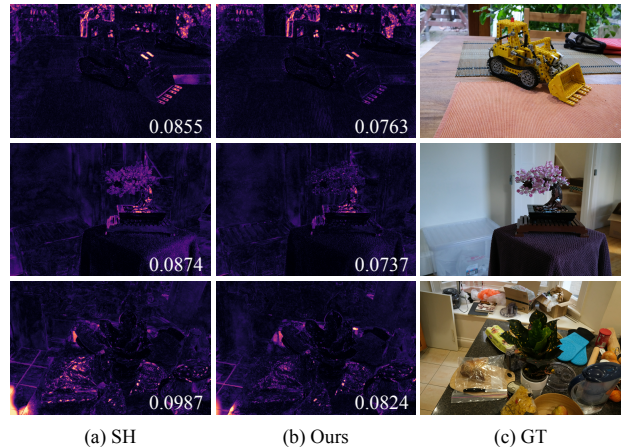


Figure 8: FLIP maps [ANAM*20] showing the error per pixel for (a) SH, (b) NASGabor and (c) GT (lower is better).

8. Conclusion

We presented a comprehensive analysis of view-dependent appearance models in the context of novel view synthesis. Based on our findings, we introduced a novel spherical kernel that incorporates the properties we identified as key for accurate modeling of view-dependent effects in the task. We demonstrated that our approach achieves strong expressiveness while maintaining a compact representation and efficient evaluation, outperforming previous works in terms of speed and quality. Most importantly, our formulation naturally enables the separation of diffuse and specular components, improving interpretability and control. We further derived analytical gradients for efficient optimization, as well as a closed-form expression for the integral of the proposed function. Beyond novel view synthesis, we believe these properties make our formulation applicable to a broader range of problems in computer graphics.

Acknowledgements

We want to thank to Wenzel Jakob for support for this project, Nicolai Hermann for valuable academic discussions, and Sophie Kergaßner for her assistance with figure design. This project has received funding from the Swiss National Science Foundation (SNSF, Grant 200502). We acknowledge access to Alps at the

Table 5: Quantitative evaluation of training and rendering efficiency on MipNerf360. *Params* indicates parameters per primitive reserved for appearance.

Method	Params	Storage ↓	Train Time ↓	FPS ↑
Spherical Harmonics	48	747.68 MB	15m50s	128
Spherical Beta [LSC*25]	15	356.04 MB	15m08s	143
Spherical Voronoi [DSRV*26]	48	747.68 MB	17m32s	130
NASGabor – 1 lobe	12	320.44 MB	13m38s	146
NASGabor – 2 lobes	21	427.25 MB	14m42s	143
NASGabor – 4 lobes	39	640.87 MB	16m36s	136

Swiss National Supercomputing Centre, Switzerland under USI's share (project ID u6).

References

- [ANAM*20] ANDERSSON P., NILSSON J., AKENINE-MÖLLER T., OSKARSSON M., ÅSTRÖM K., FAIRCHILD M. D.: Flip: A difference evaluator for alternating images. *Proceedings of the ACM in Computer Graphics and Interactive Techniques* (2020). doi:10.1145/3406183. 6, 8, 9
- [AS01] ASHIKHMIN M., SHIRLEY P.: An anisotropic phong brdf model. *Journal of Graphics Tools* (2001). doi:10.1080/10867651.2000.10487522. 2
- [BBC*23] BONET C., BERG P., COURTY N., SEPTIER F., DRUMETZ L., PHAM M.-T.: Spherical Sliced-Wasserstein. In *International Conference on Learning Representations* (2023). 5
- [BM78] BINGHAM C., MARDIA K. V.: A small circle distribution on the sphere. *Biometrika* (1978). URL: <http://www.jstor.org/stable/2335218>. 4, 13
- [BMT*21] BARRON J. T., MILDENHALL B., TANCIK M., HEDMAN P., MARTIN-BRUALLA R., SRINIVASAN P. P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021). doi:10.1109/ICCV48922.2021.00580. 2
- [BMV*22] BARRON J. T., MILDENHALL B., VERBIN D., SRINIVASAN P. P., HEDMAN P.: Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022). doi:10.1109/CVPR52688.2022.00539. 5, 7, 8, 9
- [BMV*23] BARRON J. T., MILDENHALL B., VERBIN D., SRINIVASAN P. P., HEDMAN P.: Zip-NeRF: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023). 3
- [CFHT23] CHEN Z., FUNKHOUSER T., HEDMAN P., TAGLIASACCHI A.: MobileNeRF: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023). 2
- [CHYD26] CONDOR J., HERMANN N., YURTSEVER M. A., DIDYK P.: Gabor fields: Orientation-selective level-of-detail for volume rendering. *ACM Transactions on Graphics* (2026). doi:10.1145/3811369. 7
- [CMLND*26] CONDOR J., MOENNE-LOCCOZ N., NIMIER-DAVID M., DIDYK P., GOJCIC Z., WU Q.: Neural harmonic textures for high-quality primitive based neural reconstruction, 2026. arXiv:2604.01204. 2
- [CSB*25] CONDOR J., SPEIERER S., BODE L., BOZIC A., GREEN S., DIDYK P., JARABO A.: Don't Splat your Gaussians: Volumetric Ray-Traced Primitives for Modeling and Rendering Scattering and Emissive Media. *ACM Transactions on Graphics* (2025). doi:10.1145/3711853. 2
- [CT82] COOK R. L., TORRANCE K. E.: A reflectance model for computer graphics. *ACM Transactions on Graphics* (1982). doi:10.1145/357290.357293. 2
- [DLN25] DU B., LI R. B., DU C., NGUYEN T.: Glossgau: Efficient inverse rendering for glossy surface with anisotropic spherical gaussian, 2025. arXiv:2502.14129. 2
- [DSRV*26] DI SARIO F., REBAIN D., VERBIN D., GRANGETTO M., TAGLIASACCHI A.: Spherical voronoi: Directional appearance as a differentiable partition of the sphere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2026). 2, 4, 7, 8, 10
- [Fis53] FISHER R. A.: Dispersion on a sphere. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* (1953). doi:10.1098/rspa.1953.0064. 3, 4, 5, 13
- [GKD07] GREEN P., KAUTZ J., DURAND F.: Efficient Reflectance and Visibility Approximations for Environment Map Rendering. *Computer Graphics Forum* (2007). doi:10.1111/j.1467-8659.2007.01072.x. 2, 3
- [GKMD06] GREEN P., KAUTZ J., MATUSIK W., DURAND F.: View-dependent precomputed light transport using nonlinear gaussian function approximations. In *Proceedings of the Symposium on Interactive 3D Graphics and Games* (2006). doi:10.1145/1111411.1111413. 2, 3
- [Glo10] GLOTOT X.: Understanding the difficulty of training deep feed-forward neural networks. *Proceedings of the Journal of Machine Learning Research* (2010). 5
- [HCD25] HERMANN N., CONDOR J., DIDYK P.: Puzzle similarity: A perceptually-guided cross-reference metric for artifact detection in 3d scene reconstructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2025). 9
- [HDHN16] HEITZ E., DUPUY J., HILL S., NEUBELT D.: Real-time polygonal-light shading with linearly transformed cosines. *ACM Transactions on Graphics* (2016). doi:10.1145/2897824.2925895. 2, 3, 4, 13
- [HIT*24] HUANG J., IZUKA A., TANAKA H., KOMURA T., KITAMURA Y.: Online Neural Path Guiding with Normalized Anisotropic Spherical Gaussians. *ACM Transactions on Graphics* (2024). doi:10.1145/3649310. 2, 3, 4, 5, 6, 7, 13, 16, 17, 18
- [HPP*18] HEDMAN P., PHILIP J., PRICE T., FRAHM J.-M., DRETTAKIS G., BROSTOW G.: Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics* (2018). URL: <http://www-sop.inria.fr/reves/Basilic/2018/HPPFDB18>. 2, 7, 8, 9
- [HSRG07] HAN C., SUN B., RAMAMOORTHY R., GRINSPUN E.: Frequency domain normal map filtering. *ACM Transactions on Graphics* (2007). doi:10.1145/1276377.1276412. 2, 3
- [HYC*24] HUANG B., YU Z., CHEN A., GEIGER A., GAO S.: 2D Gaussian splatting for geometrically accurate radiance fields. In *Proceedings of the ACM SIGGRAPH* (2024). doi:10.1145/3641519.3657428. 8

- [HZRS15] HE K., ZHANG X., REN S., SUN J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2015). doi:10.1109/ICCV.2015.123. 5
- [IFDN12] IWASAKI K., FURUYA W., DOBASHI Y., NISHITA T.: Real-time rendering of dynamic scenes under all-frequency lighting using integral spherical gaussian. *Computer Graphics Forum* (2012). doi:10.1111/j.1467-8659.2012.03052.x. 2, 3
- [IKK*24] ISACK H., KAR A., KHERADMAND S., REBAIN D., SHARMA G., SUN W., TAGLIASACCHI A., TSENG Y.-C., YI K.: 3d gaussian splatting as markov chain monte carlo. In *Proceedings of the International Conference on Neural Information Processing Systems* (2024). doi:10.52202/079017-2573. 7, 8
- [IS15] IOFFE S., SZEGEDY C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning* (2015). 5
- [KHDN22] KT A., HEITZ E., DUPUY J., NARAYANAN P. J.: Bringing linearly transformed cosines to anisotropic ggx. *Proceedings of the ACM in Computer Graphics and Interactive Techniques* (2022). doi:10.1145/3522612. 2, 3
- [KKLD23] KERBL B., KOPANAS G., LEIMKUEHLER T., DRETTAKIS G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* (2023). doi:10.1145/3592433. 2, 3, 7, 16
- [KLR*22] KOPANAS G., LEIMKÜHLER T., RAINER G., JAMBON C., DRETTAKIS G.: Neural point catacaustics for novel-view synthesis of reflections. *ACM Transactions on Graphics* (2022). doi:10.1145/3550454.3555497. 2
- [KM20] KATO S., MCCULLAGH P.: Some properties of a Cauchy family on the sphere derived from the Möbius transformations. *Bernoulli* (2020). doi:10.3150/20-BEJ1222. 3, 4, 5, 13
- [KMK*24] KERBL B., MEULEMAN A., KOPANAS G., WIMMER M., LANVIN A., DRETTAKIS G.: A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics* (2024). URL: <https://repo-sam.inria.fr/fungraph/hierarchical-3d-gaussians/>. 2
- [KPZK17] KNAPITSCH A., PARK J., ZHOU Q.-Y., KOLTUN V.: Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics* (2017). 7, 8, 9
- [LBOM12] LECUN Y. A., BOTTOU L., ORR G. B., MÜLLER K.-R.: *Efficient BackProp*. Springer Berlin Heidelberg, 2012, pp. 9–48. doi:10.1007/978-3-642-35289-8_3. 5
- [LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques* (1996). doi:10.1145/237170.237199. 2
- [LHG*25] LAI S., HUANG L., GUO J., CHENG K., PAN B., LONG X., LYU J., LV C., GUO Y.: Glossygs: Inverse rendering of glossy objects with 3d gaussian splatting. *IEEE Transactions on Visualization and Computer Graphics* (2025). doi:10.1109/TVCG.2025.3547063. 2
- [LSC*25] LIU R., SUN D., CHEN M., WANG Y., FENG A.: Deformable beta splatting. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Papers* (2025). doi:10.1145/3721238.3730716. 2, 4, 7, 8, 10
- [LSS*21] LOMBARDI S., SIMON T., SCHWARTZ G., ZOLLHOEFER M., SHEIKH Y., SARAGIH J.: Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics* (2021). doi:10.1145/3450626.3459863. 2, 3
- [LWDB10] LAURIJSSSEN J., WANG R., DUTRÉ P., BROWN B.: Fast estimation and rendering of indirect highlights. *Computer Graphics Forum* (2010). doi:10.1111/j.1467-8659.2010.01726.x. 2, 3
- [MBRS*21] MARTIN-BRUALLA R., RADWAN N., SAJJADI M. S. M., BARRON J. T., DOSOVITSKIY A., DUCKWORTH D.: NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021). 2
- [MDC*21] MANTIUK R. K., DENES G., CHAPIRO A., KAPLAYAN A., RUFO G., BACHY R., LIAN T., PATNEY A.: Fovvideovdp: a visible difference predictor for wide field-of-view video. *ACM Transactions on Graphics* (2021). doi:10.1145/3450626.3459831. 6
- [MESK22] MÜLLER T., EVANS A., SCHIED C., KELLER A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics* (2022). 2
- [MG20] MOGHIMBEYGI M., GOLALIZADEH M.: Spherical Logistic Distribution. *Communications in Mathematics and Statistics* (2020). doi:10.1007/s40304-018-00171-2. 3, 4, 13
- [MGN17] MÜLLER T., GROSS M., NOVÁK J.: Practical path guiding for efficient light-transport simulation. *Computer Graphics Forum* (2017). doi:10.1111/cgf.13227. 3
- [MHK*26] MAI A., HEDSTROM T., KOPANAS G., KONTKANEN J., KUESTER F., BARRON J. T.: Radiance meshes for volumetric reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2026). 2
- [MHMB*21] MILDENHALL B., HEDMAN P., MARTIN-BRUALLA R., SRINIVASAN P. P., BARRON J. T.: Nerf in the dark: High dynamic range view synthesis from noisy raw images. *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021). 9
- [MLMP*24] MOENNE-LOCCOZ N., MIRZAEI A., PEREL O., DE LUTIO R., ESTURO J. M., STATE G., FIDLER S., SHARP N., GOJCIC Z.: 3d gaussian ray tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics* (2024). 2
- [MST*21] MILDENHALL B., SRINIVASAN P. P., TANCİK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* (2021). doi:10.1145/3503250. 2, 3
- [MV25] MANOUSSAKIS G., VAPEAS P.: Spherical harmonics and gravity intensity modeling related to a special class of triaxial ellipsoids. *Mathematics* (2025). doi:10.3390/math13132115. 3
- [NDM05] NGAN A., DURAND F., MATUSIK W.: Experimental analysis of brdf models. In *Proceedings of the Eurographics Conference on Rendering Techniques* (2005). 2
- [NLC02] NA H.-S., LEE C.-N., CHEONG O.: Voronoi diagrams on the sphere. *Computational Geometry* (2002). doi:https://doi.org/10.1016/S0925-7721(02)00077-9. 2
- [NRH03] NG R., RAMAMOORTHI R., HANRAHAN P.: All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics* (2003). doi:10.1145/882262.882280. 2
- [Pho98] PHONG B. T.: *Illumination for computer generated pictures*. Association for Computing Machinery, 1998, p. 95–101. URL: <https://doi.org/10.1145/280811.280980>. 2
- [PSCS*12] PACANOWSKI R., SALAZAR CELIS O., SCHLICK C., GRANIER X., POULIN P.: Rational brdf. *IEEE Transactions on Visualization and Computer Graphics* (2012). doi:10.1109/TVCG.2012.73. 2
- [Ram06] RAMAMOORTHI R.: *Modeling Illumination Variation with Spherical Harmonics*. Academic Press, 2006, pp. 385–424. 2
- [RBS*11] ROUSIERS C., BOUSSEAU A., SUBR K., HOLZSCHUCH N., RAMAMOORTHI R.: Real-time rendering of rough refraction. *IEEE Transactions on Visualization and Computer Graphics* (2011). doi:10.1109/TVCG.2011.282. 2, 3
- [RPFGLM23] RODRIGUEZ-PARDO C., FABRE J., GARCES E., LOPEZ-MORENO J.: Nenv: Neural environment maps for global illumination. In *Computer Graphics Forum (Proceedings of the Eurographics Conference)* (2023). doi:10.1111/cgf.14883. 2, 3
- [RPLG21] REISER C., PENG S., LIAO Y., GEIGER A.: KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs .

- In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021). doi:10.1109/ICCV48922.2021.01407.2
- [RWS*06] REN Z., WANG R., SNYDER J., ZHOU K., LIU X., SUN B., SLOAN P.-P., BAO H., PENG Q., GUO B.: Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Transactions on Graphics* (2006). doi:10.1145/1141911.1141982.2
- [Sch12] SCHRAUDOLPH N. N.: *Centering Neural Network Gradient Factors*. Springer Berlin Heidelberg, 2012, pp. 205–223. doi:10.1007/978-3-642-35289-8_14.5
- [SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics* (2002). doi:10.1145/566654.566612.2,3
- [SSIN25] SILVENNOINEN A., SLOAN P.-P., IWANICKI M., NOWROUZEZHRAI D.: Spherical Harmonic Exponentials for Efficient Glossy Reflections. *Computer Graphics Forum* (2025). doi:10.1111/cgf.70219.3
- [STC*22] SARA FRIDOVICH-KEIL AND ALEX YU, TANCİK M., CHEN Q., RECHT B., KANAZAWA A.: Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022). 2
- [SZW19] SITZMANN V., ZOLLHÖFER M., WETZSTEIN G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems* (2019). 2
- [Tre96] TRENKLER G.: Continuous univariate distributions: N.I. johnson, s. kotz & n. balakrishnan (1995): Vol. 2, 2nd edition. john wiley, new york, isbn: [pound sign] 70.00, pp xix + 719. *Computational Statistics & Data Analysis* (1996). URL: <https://EconPapers.repec.org/RePEc:eee:csdana:v:22:y:1996:i:5:p:568-569>. 2, 4, 13, 16
- [TS67] TORRANCE K., SPARROW E.: Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America* (1967). doi:10.1364/JOSA.57.001105.2
- [VADWG15] VELÁZQUEZ-ARMENDÁRIZ E., DONG Z., WALTER B., GREENBERG D. P.: Complex luminaires: Illumination and appearance rendering. *ACM Transactions on Graphics* (2015). doi:10.1145/2714571.2,3
- [War92] WARD G. J.: Measuring and modeling anisotropic reflection. *SIGGRAPH Computer Graphics* (1992). doi:10.1145/142920.134078.2
- [WHM13] WEDI N., HAMRUD M., MOZDZYNSKI G.: A fast spherical harmonics transform for global nwp and climate models. *Monthly Weather Review* (2013). doi:10.1175/MWR-D-13-00016.1.3
- [WMEM*25] WU Q., MARTINEZ ESTURO J., MIRZAEI A., MOENNE-LOCCOZ N., GOJCIC Z.: 3dgut: Enabling distorted cameras and secondary rays in gaussian splatting. *Conference on Computer Vision and Pattern Recognition* (2025). 8
- [WMLT07] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet models for refraction through rough surfaces. In *Proceedings of the Eurographics Conference on Rendering Techniques* (2007). 2
- [WSCH26] WU C., SHI X., CHU Z., HUANG Y.: Ars-gs: Anisotropic reflective spherical 3d gaussian splatting. *Journal of Imaging* (2026). doi:10.3390/jimaging12040170.2
- [WWK*24] WANG C., WOLSKI K., KERBL B., SERRANO A., BEMANA M., SEIDEL H.-P., MYSZKOWSKI K., LEIMKÜHLER T.: Cinematic gaussians: Real-time hdr radiance fields with depth. In *Computer Graphics Forum* (2024). 9
- [WYF*24] WU G., YI T., FANG J., XIE L., ZHANG X., WEI W., LIU W., TIAN Q., WANG X.: 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024). 2
- [WZT*08] WANG J., ZHAO S., TONG X., SNYDER J., GUO B.: Modeling anisotropic surface reflectance with example-based microfacet synthesis. *ACM Transactions on Graphics* (2008). 2
- [WZZ24] WURSTER S., ZHANG R., ZHENG C.: Gabor splatting for high-quality gigapixel image representations. In *ACM SIGGRAPH 2024 Posters* (2024). doi:10.1145/3641234.3671081.7
- [XSD*13] XU K., SUN W.-L., DONG Z., ZHAO D.-Y., WU R.-D., HU S.-M.: Anisotropic spherical Gaussians. *ACM Transactions on Graphics* (2013). doi:10.1145/2508363.2508386.2,3,4,13
- [YGS*24] YANG Z., GAO X., SUN Y.-T., HUANG Y.-H., LYU X., ZHOU W., JIAO S., QI X., JIN X.: Spec-gaussian: anisotropic view-dependent appearance for 3d gaussian splatting. In *Proceedings of the International Conference on Neural Information Processing Systems* (2024). 2,9
- [YHR*23] YARIV L., HEDMAN P., REISER C., VERBIN D., SRINIVASAN P. P., SZELISKI R., BARRON J. T., MILDENHALL B.: Baked sdf: Meshing neural sdfs for real-time view synthesis. In *ACM SIGGRAPH Conference Proceedings* (2023). doi:10.1145/3588432.3591536.2
- [YLK*25] YE V., LI R., KERR J., TURKULAINEN M., YI B., PAN Z., SEISKARI O., YE J., HU J., TANCİK M., KANAZAWA A.: gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research* (2025). 4,7
- [YLT*21] YU A., LI R., TANCİK M., LI H., NG R., KANAZAWA A.: Plenotrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021). 2
- [Yua20] YUAN T.: The 8-parameter fisher–bingham distribution on the sphere. *Computational Statistics* (2020). doi:10.1007/s00180-020-01023-w.3,4,5,6,13
- [YZXW12] YAN L.-Q., ZHOU Y., XU K., WANG R.: Accurate translucent material rendering under spherical gaussian lights. *Computer Graphics Forum* (2012). doi:10.1111/j.1467-8659.2012.03220.x.2,3
- [ZHD*25] ZHOU J., HUANG Y., DAI W., ZOU J., ZHENG Z., KAN N., LI C., XIONG H.: 3dgabsplat: 3d gabor splatting for frequency-adaptive radiance field rendering. In *Proceedings of the ACM International Conference on Multimedia* (2025). doi:10.1145/3746027.3754855.7
- [ZNM*26] ZHOU X., NGUYEN B.-H., MAGNE L., GOLYANIK V., LEIMKÜHLER T., THEOBALT C.: Splat the net: Radiance fields with splatting neural primitives. In *International Conference on Learning Representations* (2026). 2
- [ZWZW25] ZHANG W., WANG T., ZHAN Z., WANG X.: Glossy-gaussian: Adaptive anisotropic gaussians for view-dependent appearances. *IEEE Signal Processing Letters* (2025). doi:10.1109/LSP.2025.3540372.2,9

Supplementary Material A

Distribution	PDF	Normalization Constant
Spherical Gaussian (SG) [Fis53]	$G(\mathbf{d}; \mathbf{z}, \lambda) = \exp\{\lambda(\mathbf{z} \cdot \mathbf{d} - 1)\}$	$c = \frac{\lambda}{2\pi \cdot (1 - e^{-2\lambda})}$
Spherical Cauchy [KM20]	$G(\mathbf{d}; \mathbf{z}, \rho) = \left(\frac{1 - \rho^2}{1 - \rho^2 - 2\rho \mathbf{z} \cdot \mathbf{d}} \right)^2$	$c = \frac{1}{4\pi}$
Spherical Beta (SB) [Tre96]	$G(\mathbf{d}; \mathbf{z}, \beta) = (\mathbf{z} \cdot \mathbf{d})^\beta$	$c = \frac{\beta+1}{2\pi}$
Spherical Logistic [MG20]	$G(\mathbf{d}; \mathbf{z}, \kappa, b) = \frac{\exp(k \mathbf{z} \cdot \mathbf{d})}{(b-1 + \exp(k \mathbf{z} \cdot \mathbf{d}))^2}$	$c = \frac{k(b^2 + 2(b-1)(\cosh k - 1))}{4\pi \sinh k}$
Fisher Bingham 4-param (FB4) [BM78]	$G(\mathbf{d}; \mathbf{z}, \kappa, b) = \exp\{-\kappa(\mathbf{z} \cdot \mathbf{d} - b)^2\}$	$c = \sqrt{\kappa} [2\pi^{3/2} (\Phi(\sqrt{2\kappa}(1-b)) - \Phi(\sqrt{2\kappa}(1+b)))]^{-1}$, $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt$
Fisher Bingham 6-param (FB6) [Yua20]	$G(\mathbf{d}; [\mathbf{x}, \mathbf{y}, \mathbf{z}], \kappa, \beta, \eta) = \exp(\kappa \mathbf{z} \cdot \mathbf{d} + \beta[(\mathbf{x} \cdot \mathbf{d})^2 - \eta(\mathbf{y} \cdot \mathbf{d})^2])$	$c \approx \begin{cases} 2\pi \exp\left(\beta\left(1 + \frac{\kappa^2}{4\beta^2}\right)\right) {}_1F_1\left(0.5, 1, \beta(1+\eta)\left(\frac{\kappa^2}{4\beta^2} - 1\right)\right) \sqrt{\frac{\kappa}{\beta}} & \text{if } \kappa < 2\beta \\ 2\pi e^{\kappa} [(\kappa - 2\beta)(\kappa + 2\beta\eta)]^{-\frac{1}{2}} & \text{if } 2\beta < \kappa \end{cases}$
Fisher Bingham 8-param (FB8) [Yua20]	$G(\mathbf{d}; [\mathbf{x}, \mathbf{y}, \mathbf{z}], \kappa, \beta, \eta, \mu) = \exp(\kappa \mu^\top R \mathbf{d} + \beta[(\mathbf{x} \cdot \mathbf{d})^2 - \eta(\mathbf{y} \cdot \mathbf{d})^2])$	Not analytical
Anisotropic Spherical Gaussian (ASG) [XSD* 13]	$G(\mathbf{d}; \mathbf{x}, \mathbf{y}, \mathbf{z}, \mu, \lambda, c) = \max(\mathbf{d} \cdot \mathbf{z}, 0) \exp\{-\mu(\mathbf{d} \cdot \mathbf{x})^2 - \lambda(\mathbf{d} \cdot \mathbf{y})^2\}$	Not analytical
NASG [HIT* 24]	$G(\mathbf{d}; [\mathbf{x}, \mathbf{y}, \mathbf{z}], \lambda, a) = \begin{cases} e^{\{2\lambda\kappa^{1+\tau} - 2\lambda\}} \kappa^\tau & \text{if } \mathbf{d} \neq \pm \mathbf{z}, \\ 1 & \text{if } \mathbf{d} = \mathbf{z}, \\ 0 & \text{if } \mathbf{d} = -\mathbf{z}. \end{cases}$ $\kappa = \frac{\mathbf{d} \cdot \mathbf{z} + 1}{2}, \quad \tau = \frac{a(\mathbf{d} \cdot \mathbf{x})^2}{1 - (\mathbf{d} \cdot \mathbf{z})^2}$	$c = \frac{2\pi(1 - e^{-2\lambda})}{\lambda \sqrt{1+a}}$
Linearly Transformed Cosine (LTC) [HDHN16]	$D(\mathbf{d}; M) = \max\left(0, \left(\frac{M^{-1}\mathbf{d}}{\ M^{-1}\mathbf{d}\ }\right)_z\right) \frac{ \det(M^{-1}) }{\ M^{-1}\mathbf{d}\ ^3}$	$c = \frac{1}{\pi}$

Table 6: Spherical distributions presented in our experiment with their PDFs and normalization constants, if available.

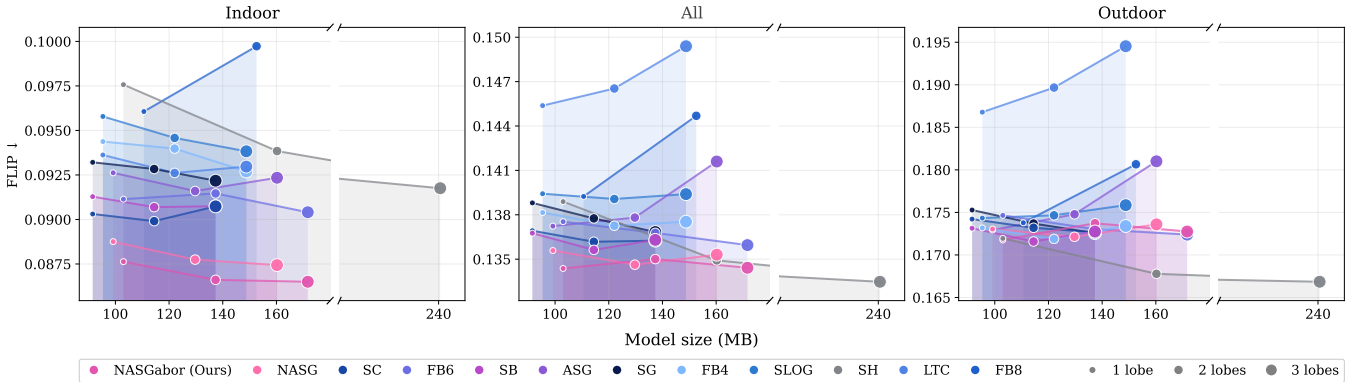
Supplementary Material B

Metrics

We include a complete breakdown of the performance of our method, per individual scene, and for a variable number of lobes, in Table 7.

Table 7: Per-scene evaluation of NASGabor with varying number of lobes across all datasets.

Scene	1 Lobe			2 Lobes			3 Lobes			4 Lobes			5 Lobes		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Mip-NeRF360															
bicycle	25.24	0.7795	0.1773	25.23	0.7809	0.1755	25.31	0.7822	0.1754	25.29	0.7822	0.1753	25.29	0.7823	0.1760
bonsai	34.49	0.9554	0.1841	34.60	0.9560	0.1835	34.68	0.9564	0.1826	34.78	0.9566	0.1824	34.81	0.9566	0.1824
counter	30.76	0.9267	0.1723	31.03	0.9282	0.1701	31.01	0.9288	0.1693	31.00	0.9289	0.1688	31.15	0.9292	0.1684
flowers	21.84	0.6259	0.3060	21.81	0.6271	0.3045	21.82	0.6263	0.3067	21.82	0.6275	0.3047	21.80	0.6272	0.3056
garden	27.43	0.8674	0.1061	27.50	0.8687	0.1049	27.46	0.8689	0.1047	27.47	0.8693	0.1040	27.51	0.8697	0.1039
kitchen	32.82	0.9351	0.1204	32.98	0.9359	0.1195	32.92	0.9361	0.1189	32.98	0.9363	0.1186	33.00	0.9365	0.1184
room	33.32	0.9375	0.1890	33.42	0.9378	0.1887	33.41	0.9371	0.1891	33.27	0.9355	0.1894	33.54	0.9379	0.1888
stump	26.92	0.7916	0.1901	26.78	0.7906	0.1905	26.78	0.7886	0.1922	26.76	0.7886	0.1926	26.74	0.7887	0.1924
treehill	22.77	0.6451	0.2921	22.84	0.6466	0.2917	22.89	0.6462	0.2908	22.79	0.6465	0.2921	22.78	0.6472	0.2939
Mean	28.40	0.8294	0.1930	28.46	0.8302	0.1921	28.48	0.8301	0.1922	28.46	0.8302	0.1920	28.51	0.8306	0.1922
NeRF-Synthetic															
chair	36.21	0.9885	0.0110	36.41	0.9889	0.0104	36.54	0.9891	0.0101	36.63	0.9892	0.0099	36.66	0.9892	0.0098
drums	26.21	0.9537	0.0368	26.48	0.9548	0.0352	26.58	0.9554	0.0343	26.67	0.9556	0.0337	26.72	0.9559	0.0335
ficus	34.04	0.9861	0.0130	34.77	0.9875	0.0118	35.33	0.9885	0.0109	35.67	0.9890	0.0104	35.88	0.9894	0.0101
hotdog	37.72	0.9870	0.0157	37.96	0.9874	0.0149	38.04	0.9875	0.0148	38.06	0.9876	0.0146	38.02	0.9874	0.0147
lego	36.45	0.9851	0.0132	36.55	0.9853	0.0129	36.60	0.9853	0.0128	36.65	0.9853	0.0128	36.67	0.9853	0.0127
materials	29.35	0.9604	0.0346	29.92	0.9628	0.0324	30.22	0.9639	0.0314	30.39	0.9645	0.0307	30.48	0.9648	0.0303
mic	35.66	0.9920	0.0057	36.15	0.9926	0.0053	36.29	0.9929	0.0051	36.45	0.9931	0.0050	36.55	0.9932	0.0049
ship	30.70	0.9066	0.1032	30.95	0.9076	0.0997	30.92	0.9076	0.0976	31.10	0.9077	0.0969	31.08	0.9077	0.0957
Mean	33.29	0.9699	0.0291	33.65	0.9709	0.0278	33.81	0.9713	0.0271	33.95	0.9715	0.0268	34.01	0.9716	0.0265
Tanks & Temples															
train	22.60	0.8329	0.1869	22.61	0.8357	0.1832	22.62	0.8332	0.1834	22.84	0.8357	0.1830	22.73	0.8350	0.1821
truck	26.68	0.9020	0.1058	26.75	0.9026	0.1058	26.69	0.9021	0.1064	26.74	0.9024	0.1057	26.73	0.9022	0.1059
Mean	24.64	0.8674	0.1463	24.68	0.8692	0.1445	24.66	0.8677	0.1449	24.79	0.8691	0.1444	24.73	0.8686	0.1440
DeepBlending															
drjohnson	30.21	0.9131	0.2271	30.15	0.9127	0.2281	29.86	0.9095	0.2291	29.88	0.9102	0.2297	29.60	0.9084	0.2310
playroom	29.61	0.9091	0.2373	30.32	0.9172	0.2291	30.39	0.9156	0.2330	30.86	0.9180	0.2299	30.19	0.9106	0.2319
Mean	29.91	0.9111	0.2322	30.24	0.9149	0.2286	30.13	0.9125	0.2311	30.37	0.9141	0.2298	29.90	0.9095	0.2315

**Figure 9:** Comparison between reconstruction quality (FLIP) and model memory for different spherical functions with varying number of lobes, grouped by scene type.

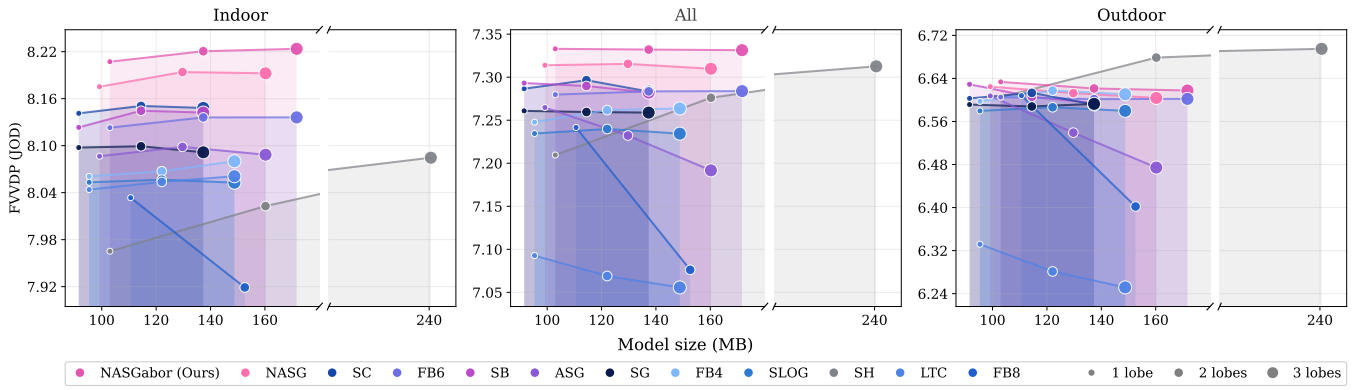


Figure 10: Comparison between reconstruction quality (FVVDP) and model memory for different spherical functions with varying number of lobes, grouped by scene type.

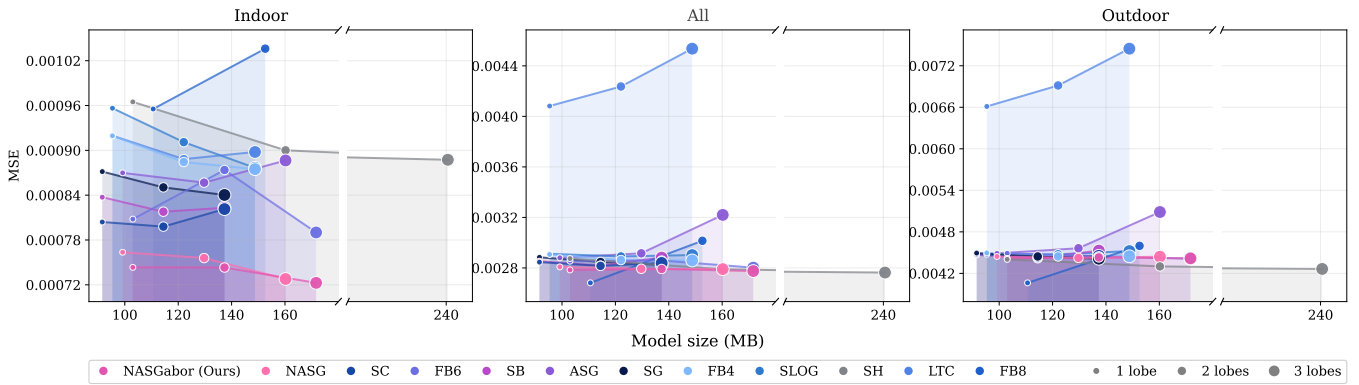


Figure 11: Comparison between reconstruction quality (MSE) and model memory for different spherical functions with varying number of lobes, grouped by scene type.

Supplementary Material C

Kernel parametrization

Parameter	Activation	Range
$\cos \theta, \cos \phi, \cos \tau$	$\tanh(x)$	$[-1, 1]$
λ	$\exp(x)$	$(0, \infty)$
a	$\exp(x)$	$(0, \infty)$
k	$20 \cdot [\tanh(x) + 1]$	$[0, 40]$
Weights	$\tanh(x)$	$[-1, 1]$

Table 8: Activation functions and valid ranges for the NASGabor parameterization. The frequency parameter k is bounded to prevent high-frequency oscillations that cannot be reliably resolved from the input views, which would otherwise lead to aliasing artifacts.

Lr scheduling

We use a cosine scheduler to tune appearance model learning rate throughout the optimization, following

$$\eta(t) = \frac{1}{2} \left(1 + \cos \left(\frac{\pi t}{T_{\max}} \right) \right), \quad (11)$$

where T_{\max} denotes the total number of training iterations and t the current iteration. The schedule is applied after the model has largely converged (after 7k iterations), which helps stabilize late-stage optimization. We apply this scheduling to all learning rates except for l_{xyz} , which follows its own schedule as introduced in [KKLD23].

Parameterization for NASG and NASGabor

Originally, the work that introduced NASG [HIT*24] distributions, in the context of neural path guiding, proposed a 7-parameter kernel. It uses an over-parameterized Euler orthogonal basis, for which an MLP predicts, in some cases, both the sine and cosine of the same angles. While this may be desirable in neural rendering settings, such a parameterization is extremely unstable when attempting to regress mixture models of the kernel itself directly, without an MLP decoder, as independently learning the sine and cosine of the same angle is not guaranteed to be consistent and represents a waste of parameters. Instead, we propose a slightly different parameterization based on the same orthogonal basis idea. The parameterization uses 2 parameters for the mean direction on the unit sphere, 2 parameters for the covariance (the anisotropic “shape” of the kernel), and 1 parameter for the in-plane rotation around the mean direction. This reduces the total parameter count to 5 and allows direct optimization without requiring an MLP decoder.

More concretely, we construct the orthonormal frame to translate and rotate our distribution over the surface of the sphere by optimizing the cosine terms of

$$\cos \theta = c_\theta, \quad \cos \phi = c_\phi, \quad \cos \tau = c_\tau,$$

with corresponding sine terms defined as

$$\sin \theta = s_\theta = \sqrt{1 - c_\theta^2}, \quad \sin \phi = s_\phi = \sqrt{1 - c_\phi^2},$$

$$\sin \tau = s_\tau = \sqrt{1 - c_\tau^2}.$$

The kernel mean direction, corresponding to the local z -axis, is parameterized on the unit sphere as

$$\mathbf{z} = \begin{bmatrix} c_\theta s_\phi \\ s_\theta s_\phi \\ c_\phi \end{bmatrix}.$$

A tangent direction \mathbf{x} , rotated around \mathbf{z} by an angle τ , is constructed as

$$\mathbf{x} = \begin{bmatrix} c_\theta c_\phi c_\tau - s_\theta s_\tau \\ s_\theta c_\phi c_\tau + c_\theta s_\tau \\ -s_\phi c_\tau \end{bmatrix}.$$

The remaining orthogonal tangent vector is then obtained through the cross product

$$\mathbf{y} = \mathbf{z} \times \mathbf{x}.$$

Together, the orthonormal frame is given by

$$\mathbf{R} = [\mathbf{x} \quad \mathbf{y} \quad \mathbf{z}].$$

This construction defines a minimal orthonormal frame on the sphere, where (θ, ϕ) parameterize the mean direction and τ defines the rotational orientation in the tangent plane.

Supplementary Material D

Normalization of Spherical β -distribution [Tre96]

Let $\mathbf{z}, \mathbf{d} \in \mathbb{S}^2$ be unit vectors and define

$$\mu = \mathbf{z} \cdot \mathbf{d} = \cos \theta,$$

where θ is the angle between \mathbf{z} - the direction of the highest distribution peak and \mathbf{d} - viewing direction. The kernel supported on the hemisphere is defined as

$$G_\beta(\mathbf{d}; \mathbf{z}, \beta) = \begin{cases} (\mathbf{z} \cdot \mathbf{d})^\beta = \mu^\beta = (\cos \theta)^\beta, & \text{if } \mu \in [0, 1], \\ 0, & \text{otherwise.} \end{cases}$$

We normalize the kernel over the hemisphere, where the value of the distribution is different than 0.

$$\mathbb{S}_+^2 = \{\mathbf{d} \in \mathbb{S}^2 : \mathbf{z} \cdot \mathbf{d} \geq 0\}.$$

Using spherical coordinates with $\theta \in [0, \frac{\pi}{2}]$ and $\phi \in [0, 2\pi]$, and surface element $d\omega = \sin \theta d\theta d\phi$, we obtain

$$C_\beta = \int_{\mathbb{S}_+^2} (\mathbf{z} \cdot \mathbf{d})^\beta d\omega = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} (\cos \theta)^\beta \sin \theta d\theta d\phi.$$

With the substitution $x = \cos \theta$, this becomes

$$C_\beta = 2\pi \int_0^1 x^\beta dx = \frac{2\pi}{\beta + 1}.$$

The normalized kernel is therefore

$$G_\beta(\mathbf{d}; \mathbf{z}, \beta) = \begin{cases} \frac{\beta + 1}{2\pi} (\mathbf{z} \cdot \mathbf{d})^\beta, & \text{if } \mathbf{z} \cdot \mathbf{d} \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

Supplementary Material E

NASG Derivatives

We present the analytical derivatives of the NASG [HIT*24] distribution with respect to its unactivated parameters a , λ , \mathbf{z} , and \mathbf{x} .

For

$$G(\mathbf{d}; [\mathbf{x}, \mathbf{y}, \mathbf{z}], \lambda, a) =$$

$$\begin{cases} e \left(2\lambda \left(\frac{\mathbf{d} \cdot \mathbf{z} + 1}{2} \right)^{1 + \frac{a(\mathbf{d} \cdot \mathbf{x})^2}{1 - (\mathbf{d} \cdot \mathbf{z})^2}} - 2\lambda \right) \left(\frac{\mathbf{d} \cdot \mathbf{z} + 1}{2} \right)^{\frac{a(\mathbf{d} \cdot \mathbf{x})^2}{1 - (\mathbf{d} \cdot \mathbf{z})^2}} & \text{if } \mathbf{d} \neq \pm \mathbf{z}, \\ 1 & \text{if } \mathbf{d} = \mathbf{z}, \\ 0 & \text{if } \mathbf{d} = -\mathbf{z}. \end{cases} \quad (12)$$

Let

$$\kappa = \frac{\mathbf{d} \cdot \mathbf{z} + 1}{2}, \quad \tau = \frac{a(\mathbf{d} \cdot \mathbf{x})^2}{1 - (\mathbf{d} \cdot \mathbf{z})^2}, \quad E = \kappa^\tau \quad (13)$$

$$p = \exp\{2\lambda(E \cdot \kappa - 1)\}, \quad C^{-1} = \frac{\lambda\sqrt{1+a}}{2\pi(1 - e^{-2\lambda})} \quad (14)$$

Then,

$$G(\mathbf{d}; [\mathbf{x}, \mathbf{y}, \mathbf{z}], \lambda, a) = p \cdot E \cdot C^{-1} \quad (15)$$

Derivative with respect to a

The derivative with respect to the anisotropy parameter a can be computed analytically using the chain rule.

$$\frac{\partial \tau}{\partial a} = \frac{(\mathbf{d} \cdot \mathbf{x})^2}{1 - (\mathbf{d} \cdot \mathbf{z})^2} \quad (16)$$

The derivative can be split into three main components:

$$\frac{\partial p}{\partial a} = p \cdot (2\lambda E \kappa) \ln(\tau) \frac{\partial \tau}{\partial a} \quad (17)$$

$$\frac{\partial E}{\partial a} = E \ln(\kappa) \frac{\partial \tau}{\partial a} \quad (18)$$

$$\frac{\partial C^{-1}}{\partial a} = \frac{\lambda}{2\sqrt{1+a}} \cdot \frac{1}{2\pi(1 + \epsilon - e^{-2\lambda})} \quad (19)$$

Finally, the total derivative of the NASG [HIT*24] PDF with respect to a is

$$\frac{\partial G}{\partial a} = \underbrace{\frac{\partial p}{\partial a} \cdot E \cdot C^{-1}}_{\text{contribution from } p} + \underbrace{p \cdot \frac{\partial E}{\partial a} \cdot C^{-1}}_{\text{contribution from } E} + \underbrace{p \cdot E \cdot \frac{\partial C^{-1}}{\partial a}}_{\text{contribution from } C^{-1}} \quad (20)$$

Derivative with respect to λ

The derivative with respect to the scale parameter λ can be computed analytically using the product rule.

$$\frac{\partial p}{\partial \lambda} = 2p(E \cdot \kappa - 1) \quad (21)$$

$$\frac{\partial C^{-1}}{\partial \lambda} = \frac{\sqrt{1+a}(1 + \epsilon - e^{-2\lambda}) + 2\lambda\sqrt{1+a}e^{-2\lambda}}{2\pi(1 + \epsilon - e^{-2\lambda})^2} \quad (22)$$

Then, the derivative is

$$\frac{\partial G}{\partial \lambda} = \underbrace{E \cdot C^{-1} \cdot \frac{\partial p}{\partial \lambda}}_{\text{contribution from } p} + \underbrace{E \cdot p \cdot \frac{\partial C^{-1}}{\partial \lambda}}_{\text{contribution from } C^{-1}} \quad (23)$$

Derivative with respect to \mathbf{x}

The derivative with respect to the basis vector \mathbf{x} is obtained analytically using the chain rule.

$$\frac{\partial \tau}{\partial \mathbf{x}} = \frac{2a(\mathbf{d} \cdot \mathbf{x})}{1 - (\mathbf{d} \cdot \mathbf{z})^2} \mathbf{d} \quad (24)$$

Then,

$$\frac{\partial E}{\partial \mathbf{x}} = E \ln(\kappa) \frac{\partial \tau}{\partial \mathbf{x}}, \quad (25)$$

$$\frac{\partial p}{\partial \mathbf{x}} = p 2\lambda \kappa \frac{\partial E}{\partial \mathbf{x}} \quad (26)$$

Finally,

$$\frac{\partial G}{\partial \mathbf{x}} = \underbrace{\frac{\partial p}{\partial \mathbf{x}} \cdot E \cdot C^{-1}}_{\text{contribution from } p} + \underbrace{p \cdot \frac{\partial E}{\partial \mathbf{x}} \cdot C^{-1}}_{\text{contribution from } E} \quad (27)$$

Derivative with respect to \mathbf{z}

The derivative with respect to the basis vector \mathbf{z} is obtained analytically using the chain rule. The derivatives of κ and τ with respect to \mathbf{z} are

$$\frac{\partial \kappa}{\partial \mathbf{z}} = \frac{1}{2} \mathbf{d}, \quad (28)$$

$$\frac{\partial \tau}{\partial \mathbf{z}} = a(\mathbf{d} \cdot \mathbf{x})^2 \frac{2(\mathbf{d} \cdot \mathbf{z})}{(1 - (\mathbf{d} \cdot \mathbf{z})^2)^2} \mathbf{d} \quad (29)$$

Then,

$$\frac{\partial E}{\partial \mathbf{z}} = E \left[\frac{\tau}{\kappa} \frac{\partial \kappa}{\partial \mathbf{z}} + \ln(\kappa) \frac{\partial \tau}{\partial \mathbf{z}} \right] \quad (30)$$

$$\frac{\partial p}{\partial \mathbf{z}} = p 2\lambda \left(\kappa \frac{\partial E}{\partial \mathbf{z}} + E \frac{\partial \kappa}{\partial \mathbf{z}} \right) \quad (31)$$

Finally,

$$\frac{\partial G}{\partial \mathbf{z}} = \underbrace{\frac{\partial p}{\partial \mathbf{z}} \cdot E \cdot C^{-1}}_{\text{contribution from } p} + \underbrace{p \cdot \frac{\partial E}{\partial \mathbf{z}} \cdot C^{-1}}_{\text{contribution from } E} \quad (32)$$

Auxiliary derivatives and notation. Let

$$c_\theta = \cos \theta, \quad s_\theta = \sin \theta = \sqrt{1 - c_\theta^2}, \quad (33)$$

$$c_\phi = \cos \phi, \quad s_\phi = \sin \phi = \sqrt{1 - c_\phi^2}, \quad (34)$$

$$c_\psi = \cos \psi, \quad s_\psi = \sin \psi = \sqrt{1 - c_\psi^2} \quad (35)$$

The derivative of the sine with respect to the cosine is given by

$$\frac{\partial s}{\partial c} = -\frac{c}{s}, \quad (36)$$

which is used throughout the following derivations.

Derivative with respect to $\cos \theta, \cos \phi, \cos \psi$

Since the basis vectors \mathbf{x} and \mathbf{z} depend on the scalar parameters c_θ , c_ϕ , and c_ψ , the derivative of G with respect to each parameter is obtained via the chain rule.

Derivative with respect to c_θ .

$$\frac{\partial G}{\partial c_\theta} = \frac{\partial G}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial c_\theta} + \frac{\partial G}{\partial \mathbf{z}} \cdot \frac{\partial \mathbf{z}}{\partial c_\theta}. \quad (37)$$

The partial derivatives of the basis vectors are

$$\frac{\partial \mathbf{x}}{\partial c_\theta} = \begin{pmatrix} c_\phi c_\psi - \frac{\partial s_\theta}{\partial c_\theta} s_\psi \\ \frac{\partial s_\theta}{\partial c_\theta} c_\phi c_\psi + s_\psi \\ 0 \end{pmatrix} = \begin{pmatrix} c_\phi c_\psi + \frac{c_\theta}{s_\theta} s_\psi \\ -\frac{c_\theta}{s_\theta} c_\phi c_\psi + s_\psi \\ 0 \end{pmatrix}, \quad (38)$$

$$\frac{\partial \mathbf{z}}{\partial c_\theta} = \begin{pmatrix} s_\phi \\ \frac{\partial s_\theta}{\partial c_\theta} s_\phi \\ 0 \end{pmatrix} = \begin{pmatrix} s_\phi \\ \frac{c_\theta}{s_\theta} s_\phi \\ 0 \end{pmatrix} \quad (39)$$

Derivative with respect to c_ϕ .

$$\frac{\partial G}{\partial c_\phi} = \frac{\partial G}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial c_\phi} + \frac{\partial G}{\partial \mathbf{z}} \cdot \frac{\partial \mathbf{z}}{\partial c_\phi}. \quad (40)$$

$$\frac{\partial \mathbf{x}}{\partial c_\phi} = \begin{pmatrix} c_\theta c_\psi \\ s_\theta c_\psi \\ -\frac{\partial s_\phi}{\partial c_\phi} c_\psi \\ \frac{\partial s_\phi}{\partial c_\phi} c_\psi \end{pmatrix} = \begin{pmatrix} c_\theta c_\psi \\ s_\theta c_\psi \\ \frac{c_\phi}{s_\phi} c_\psi \\ -\frac{c_\phi}{s_\phi} c_\psi \end{pmatrix}, \quad (41)$$

$$\frac{\partial \mathbf{z}}{\partial c_\phi} = \begin{pmatrix} c_\theta \frac{\partial s_\phi}{\partial c_\phi} \\ c_\theta \frac{\partial s_\phi}{\partial c_\phi} \\ s_\theta \frac{\partial s_\phi}{\partial c_\phi} \\ 1 \end{pmatrix} = \begin{pmatrix} -\frac{c_\theta c_\phi}{s_\phi} \\ -\frac{s_\theta c_\phi}{s_\phi} \\ \frac{s_\phi}{1} \\ 1 \end{pmatrix} \quad (42)$$

Derivative with respect to c_ψ .

Since \mathbf{z} does not depend on τ , the derivative simplifies to

$$\frac{\partial G}{\partial c_\psi} = \frac{\partial G}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial c_\psi} \quad (43)$$

$$\frac{\partial \mathbf{x}}{\partial c_\psi} = \begin{pmatrix} c_\theta c_\phi - \frac{\partial s_\psi}{\partial c_\psi} s_\theta \\ s_\theta c_\phi + \frac{\partial s_\psi}{\partial c_\psi} c_\theta \\ -s_\phi \\ \frac{\partial s_\psi}{\partial c_\psi} c_\theta \\ -s_\phi \end{pmatrix} = \begin{pmatrix} c_\theta c_\phi + \frac{s_\theta c_\psi}{s_\psi} \\ s_\theta c_\phi - \frac{c_\theta c_\psi}{s_\psi} \\ -s_\phi \\ -s_\phi \end{pmatrix} \quad (44)$$

Supplementary Material F

Normalized Anisotropic Spherical Gabor Derivatives

In our new function, we directly reuse NASG [HIT*24] formulation. Following, the C , p and E indicates the same symbols as in

Supplementary Material E.

$$G(\mathbf{d}; [\mathbf{x}, \mathbf{y}, \mathbf{z}], \lambda, a, k) =$$

$$\begin{cases} e^{\left(2\lambda \left(\frac{\mathbf{d} \cdot \mathbf{z} + 1}{2}\right)^{1 + \frac{a(\mathbf{d} \cdot \mathbf{x})^2}{1 - (\mathbf{d} \cdot \mathbf{z})^2}} - 2\lambda\right) \left(\frac{\mathbf{d} \cdot \mathbf{z} + 1}{2}\right)^{\frac{a(\mathbf{d} \cdot \mathbf{x})^2}{1 - (\mathbf{d} \cdot \mathbf{z})^2}} \cdot \frac{1 + \cos(k \mathbf{d} \cdot \mathbf{x})}{2}} & \text{if } \mathbf{d} \neq \pm \mathbf{z}, \\ 1 & \text{if } \mathbf{d} = \mathbf{z}, \\ 0 & \text{if } \mathbf{d} = -\mathbf{z}. \end{cases} \quad (45)$$

in short:

$$G(\mathbf{d}; [\mathbf{x}, \mathbf{y}, \mathbf{z}], \lambda, a, k) = \begin{cases} p \cdot E \cdot C^{-1} \cdot \frac{1 + \cos(k \mathbf{d} \cdot \mathbf{x})}{2} & \text{if } \mathbf{d} \neq \pm \mathbf{z}, \\ 1 & \text{if } \mathbf{d} = \mathbf{z}, \\ 0 & \text{if } \mathbf{d} = -\mathbf{z}, \end{cases} \quad (46)$$

$$\kappa = \frac{\mathbf{d} \cdot \mathbf{z} + 1}{2}, \quad \tau = \frac{a(\mathbf{d} \cdot \mathbf{x})^2}{1 - (\mathbf{d} \cdot \mathbf{z})^2}, \quad E = \kappa^\tau \quad (47)$$

$$p = \exp\left(2\lambda(E \cdot \kappa - 1)\right), \quad C^{-1} = \frac{\lambda \sqrt{1+a}}{2\pi(1 - e^{-2\lambda})}. \quad (48)$$

Which allows us to rewrite the formulation as:

$$G(\mathbf{d}; [\mathbf{x}, \mathbf{y}, \mathbf{z}], \lambda, a) = p \cdot E \cdot C^{-1} \cdot \frac{1 + \cos(k \mathbf{d} \cdot \mathbf{x})}{2} \quad (49)$$

Derivative with respect to a

The derivative of the NASGabor distribution with respect to the anisotropy parameter a can be computed analytically using the chain rule. Define

$$\frac{\partial \tau}{\partial a} = \frac{(\mathbf{d} \cdot \mathbf{x})^2}{1 - (\mathbf{d} \cdot \mathbf{z})^2} \quad (50)$$

The derivative can be split into three main components:

$$\frac{\partial p}{\partial a} = p \cdot (2\lambda E \kappa) \ln(\kappa) \frac{\partial \tau}{\partial a} \quad (51)$$

$$\frac{\partial E}{\partial a} = E \ln(\kappa) \frac{\partial \tau}{\partial a} \quad (52)$$

$$\frac{\partial C^{-1}}{\partial a} = \frac{\lambda}{2\sqrt{1+a}} \cdot \frac{1}{2\pi(1 + \epsilon - e^{-2\lambda})} \quad (53)$$

Finally, the total derivative of the NASGabor PDF with respect to a is

$$\frac{\partial G}{\partial a} = \left[\underbrace{\frac{\partial p}{\partial a} E C^{-1}}_{\text{contribution from } p} + \underbrace{p \frac{\partial E}{\partial a} C^{-1}}_{\text{contribution from } E} + \underbrace{p E \frac{\partial C^{-1}}{\partial a}}_{\text{contribution from } C^{-1}} \right] \cdot \frac{1 + \cos(k \mathbf{d} \cdot \mathbf{x})}{2}$$

Derivative with respect to λ

The derivative of the NASGabor distribution with respect to the

scale parameter λ can be computed analytically using the product rule.

The derivative of each component is:

$$\frac{\partial p}{\partial \lambda} = 2p(E \cdot \kappa - 1), \quad (54)$$

$$\frac{\partial C^{-1}}{\partial \lambda} = \frac{\sqrt{1+a}(1+\epsilon - e^{-2\lambda}) + 2\lambda\sqrt{1+a}e^{-2\lambda}}{2\pi(1+\epsilon - e^{-2\lambda})^2}. \quad (55)$$

Note that E does not depend on λ , so its derivative is zero.

Finally, the total derivative of the NASGabor PDF with respect to λ is

$$\frac{\partial G}{\partial \lambda} = \left[\underbrace{E C^{-1} \frac{\partial p}{\partial \lambda}}_{\text{contribution from } p} + \underbrace{E p \frac{\partial C^{-1}}{\partial \lambda}}_{\text{contribution from } C^{-1}} \right] \cdot \frac{1 + \cos(k \mathbf{d} \cdot \mathbf{x})}{2} \quad (56)$$

Derivative with respect to \mathbf{x}

The derivative of the NASGabor distribution with respect to the basis vector \mathbf{x} can be computed analytically using the chain rule.

The derivative of τ with respect to \mathbf{x} is

$$\frac{\partial \tau}{\partial \mathbf{x}} = \frac{2a(\mathbf{d} \cdot \mathbf{x})}{1 - (\mathbf{d} \cdot \mathbf{z})^2} \mathbf{d}. \quad (57)$$

The derivatives of E and p with respect to \mathbf{x} are

$$\frac{\partial E}{\partial \mathbf{x}} = E \ln(\kappa) \frac{\partial \tau}{\partial \mathbf{x}}, \quad \frac{\partial p}{\partial \mathbf{x}} = 2\lambda p \kappa \frac{\partial E}{\partial \mathbf{x}}. \quad (58)$$

The derivative of the cosine term is

$$\frac{\partial}{\partial \mathbf{x}} \frac{1 + \cos(k \mathbf{d} \cdot \mathbf{x})}{2} = -\frac{k}{2} \sin(k \mathbf{d} \cdot \mathbf{x}) \mathbf{d} \quad (59)$$

Finally, the total derivative of the NASGabor PDF with respect to \mathbf{x} is

$$\begin{aligned} \frac{\partial G}{\partial \mathbf{x}} = & \left[\underbrace{\frac{\partial p}{\partial \mathbf{x}} \cdot E \cdot C^{-1}}_{\text{contribution from } p} + \underbrace{p \cdot \frac{\partial E}{\partial \mathbf{x}} \cdot C^{-1}}_{\text{contribution from } E} \right] \cdot \frac{1 + \cos(k \mathbf{d} \cdot \mathbf{x})}{2} \\ & + \underbrace{p E C^{-1} \frac{\partial}{\partial \mathbf{x}} \cdot \frac{1 + \cos(k \mathbf{d} \cdot \mathbf{x})}{2}}_{\text{contribution from cosine derivative}} \end{aligned} \quad (60)$$

Derivative with respect to \mathbf{z}

The derivative of the NASGabor distribution with respect to the basis vector \mathbf{z} can be computed analytically using the chain rule.

The derivatives of κ and τ with respect to \mathbf{z} are

$$\frac{\partial \kappa}{\partial \mathbf{z}} = \frac{1}{2} \mathbf{d}, \quad \frac{\partial \tau}{\partial \mathbf{z}} = a(\mathbf{d} \cdot \mathbf{x})^2 \frac{2(\mathbf{d} \cdot \mathbf{z})}{(1 - (\mathbf{d} \cdot \mathbf{z})^2)^2} \mathbf{d}. \quad (61)$$

The derivative of E with respect to \mathbf{z} is

$$\frac{\partial E}{\partial \mathbf{z}} = E \left[\frac{\tau}{\kappa} \frac{\partial \kappa}{\partial \mathbf{z}} + \ln(\kappa) \frac{\partial \tau}{\partial \mathbf{z}} \right]. \quad (62)$$

The derivative of p with respect to \mathbf{z} is

$$\frac{\partial p}{\partial \mathbf{z}} = p 2\lambda \left(\kappa \frac{\partial E}{\partial \mathbf{z}} + E \frac{\partial \kappa}{\partial \mathbf{z}} \right). \quad (63)$$

Finally, the total derivative of the NASGabor PDF with respect to \mathbf{z} is

$$\frac{\partial G}{\partial \mathbf{z}} = \left[\underbrace{\frac{\partial p}{\partial \mathbf{z}} \cdot E \cdot C^{-1}}_{\text{contribution from } p} + \underbrace{p \cdot \frac{\partial E}{\partial \mathbf{z}} \cdot C^{-1}}_{\text{contribution from } E} \right] \cdot \frac{1 + \cos(k \mathbf{d} \cdot \mathbf{x})}{2}. \quad (64)$$

Derivative with respect to k

The derivative of the NASGabor distribution with respect to the Gabor frequency k can be computed directly from the cosine term.

Since only the cosine depends on k , we have

$$\frac{\partial}{\partial k} \frac{1 + \cos(k \mathbf{d} \cdot \mathbf{x})}{2} = -\frac{(\mathbf{d} \cdot \mathbf{x})}{2} \sin(k \mathbf{d} \cdot \mathbf{x}). \quad (65)$$

Finally, the total derivative of the NASGabor PDF with respect to k is

$$\frac{\partial G}{\partial k} = p E C^{-1} \cdot \underbrace{-\frac{(\mathbf{d} \cdot \mathbf{x})}{2} \sin(k \mathbf{d} \cdot \mathbf{x})}_{\text{contribution from cosine derivative}} \quad (66)$$