Anisotropic Kernels for Neural Implicit Surface Reconstruction

Soyeon Jang[1] Adrian Ramlal[1] Paul Fieguth[1] Yuhao Chen[1]
[1]Vision and Image Processing Group, Systems Design Engineering, University of Waterloo
{soyeon.jang, avdramla, pfieguth, yuhao.chen1}@uwaterloo.ca

Abstract

Representing fuzzy and strand-like geometry such as hair, fur and other thin, directionally structured materials remains a challenge for surface-based neural rendering methods. Recent approaches provide accurate surface reconstructions but rely on isotropic parameterizations that struggle to capture directionally varying properties. We explore whether introducing directional information into these parameterizations can better model such geometry. Motivated by anisotropic kernels in 3D Gaussian Splatting, we propose two extensions that augment the kernel with either viewing direction or local surface normals. We integrate these variants into surface-based rendering pipeline and perform an evaluation on selected Shelly scenes. Our findings indicate that incorporating directional information improves reconstruction quality, with normal-aligned kernels performing best on strand-like fur and view-aligned kernels excelling on highly anisotropic geometry. While our study is limited to two scenes, the results suggest that anisotropic parameterizations merit further investigation for modeling fuzzy materials.

1. Introduction

Accurately modeling fuzzy geometry—such as hair, fur, grass, and fibrous materials—remains a fundamental challenge in neural rendering. These structures exhibit thin, directionally-oriented features with complex view-dependent appearance, making them difficult to capture using conventional representations. High-fidelity reconstruction of such geometry is crucial for realistic depiction of animals, digital humans, and materials including wool fabrics, microfiber textiles, and plant foliage.

Recent advances in neural implicit surface reconstruction have demonstrated impressive results by converting signed distance functions (SDFs) to volume density via kernel-based transformations. NeuS [9] introduced a logistic kernel with a global scalar parameter s controlling density spread, while Adaptive Shells [11] extended this to a spatially-varying scalar field $s(\mathbf{x})$, enabling local adaptation between sharp surfaces and volumetric regions. However, these approaches remain fundamentally *isotropic*,the kernel spreads uniformly in all directions at each point, regardless of the underlying geometric structure.

Motivated by the anisotropic kernels used in 3D Gaussian Splatting (3DGS) [5], we explore directionally aware alternatives to the standard position-dependent scalar parameter. Specifically, we investigate two variants of vector-valued kernels that introduce forms of anisotropy into the representation. Our main contribution is an exploratory evaluation of anisotropic kernels that incorporate simple directional cues. These variants highlight the potential benefits of directionally aware parameters for representing thin, fuzzy geometries.

2. Related Work

Neural Radiance Fields and Implicit Surfaces. Neural Radiance Fields (NeRF) [7] pioneered differentiable volume rendering of neural scene representations, directly learning density and view-dependent color from 2D images. While achieving photorealistic synthesis, NeRF's density-based representation lacks explicit surface constraints, making high-quality geometry extraction challenging. Subsequent works have extended NeRF with improved efficiency [8], anti-aliasing [1], and unbounded scene handling [2].

To enable accurate surface reconstruction, several methods represent geometry as SDFs. NeuS [9] converts an SDF to volume density using a logistic distribution with global kernel width s, unifying the benefits of surface representation with robust volume rendering optimization. VolSDF [12] employs a similar approach with Laplace distributions. These methods demonstrate superior surface quality compared to density-based NeRF, but their global kernel parameters limit adaptivity to varying geometric complexity across the scene. Recent variants like NeuS2 [10] and Neuralangelo [6] improve training efficiency and incorporate additional supervision, but maintain the fundamental SDF-to-density framework.

Spatially-Adaptive Representations. Recognizing that scenes contain both sharp surfaces and volumetric regions, recent work has explored local adaptivity. Instant NGP [8] introduced multi-resolution hash grids for efficient spatial feature encoding, dramatically accelerating training and inference. Building on this, Adaptive Shells [11] extends NeuS with a spatially-varying scalar kernel $s(\mathbf{x})$, allowing automatic adaptation between sharp surfaces (small s) and fuzzy volumetric regions (large s). This approach extracts explicit mesh shells to accelerate rendering by restricting sampling to regions near surfaces. However, the scalar kernel remains isotropic, limiting its ability to capture directional geometric features.

Anisotropic Representations. Anisotropy has proven effective in explicit 3D representations. 3DGS [5] represents scenes as collections of oriented 3D Gaussians with anisotropic covariances, enabling efficient capture of directional structure and real-time rendering through rasterization. Variants like 2D Gaussian Splatting [4] use anisotropic 2D kernels within volumetric rendering for improved efficiency. These works demonstrate that anisotropic primitives effectively capture local geometric directionality. However, they rely on explicit point-based representations rather than continuous implicit fields.

For implicit representations, Volumetric Surfaces [3] learns multiple concentric SDF shells to represent fuzzy geometry like hair and fur that single surfaces cannot capture. While this multi-layer approach handles volumetric detail, each layer still uses isotropic density conversion. Our work introduces anisotropy directly into the SDF-to-density conversion kernel, enabling directional adaptation within a continuous implicit framework. This bridges neural implicit surfaces with the directional adaptivity of Gaussian-based methods, while maintaining the advantages of SDF representations for surface reconstruction.

3. Method

3.1. Background: SDF-to-Density Conversion

Following NeuS [9], we represent the scene geometry as a signed distance function (SDF) $f : \mathbb{R}^3 \to \mathbb{R}$, which is converted to volume density σ for rendering via:

$$\sigma = \max\left(\frac{-\frac{d\Phi_s}{d\tau}(f)}{\Phi_s(f),}0\right), \quad \Phi_s(f) = (1+e^{-f/s})^{-1} \quad (1)$$

where Φ_s is the sigmoid function and s controls the "sharpness" or spread of the density around the zero-level set. NeuS learns a single global scalar s for the entire scene, while Adaptive Shells [11] extends this to a spatially-varying scalar field $s(\mathbf{x})$, allowing the representation to adapt between sharp surfaces (small s) and volumetric regions (large s).

3.2. Anisotropic Kernel Representation

While the spatially-varying scalar kernel of Adaptive Shells [11] provides local adaptivity, it remains **isotropic**, the density spread is uniform in all spatial axes at each point. However, many geometric features exhibit directional characteristics. For instance, a thin strand-like structure has vastly different extent along its normal direction compared to its tangent plane.

3.3. Directional Kernel Projection

The 3D anisotropic kernel $\mathbf{s}$ provides directional control, and we investigate two projection strategies to compute an effective scalar kernel size for density conversion:

View-Direction Projection. For volumetric rendering along a ray with direction $\mathbf{d}$, we project the anisotropic kernel onto the viewing direction:

$$s_{\text{eff}} = \mathbf{s}^T(\hat{\mathbf{d}} \odot \hat{\mathbf{d}}) = \sum_{i=1}^{3} s_i \hat{d}_i^2 \quad (2)$$

where $\hat{\mathbf{d}} = \mathbf{d}/\|\mathbf{d}\|$ is the normalized ray direction and $\odot$ denotes element-wise multiplication. This formulation ensures that the effective kernel size reflects the anisotropy along the ray direction, adapting the density spread based on how the ray intersects local geometric features.

Normal-Direction Projection. Alternatively, surface geometry is fundamentally characterized by its normal direction. We project the kernel onto the surface normal $\mathbf{n} = \nabla f/\|\nabla f\|$:

$$s_{\text{eff}} = \mathbf{s}^T(\hat{\mathbf{n}} \odot \hat{\mathbf{n}}) = \sum_{i=1}^{3} s_i \hat{n}_i^2 \quad (3)$$

This approach ties the kernel anisotropy to the geometric structure itself rather than the viewing configuration, potentially providing more stable and geometrically meaningful adaption.

3.4. Volume Rendering with Anisotropic Kernels

During rendering, we compute sample-wise densities using the effective kernel size s_{eff} computed at interval midpoints. For view-direction projection, this is incorporated into the NeuS volume rendering formulation [9] as:

$$\alpha_i = \max\left(\frac{\Phi_{s_{\text{eff},i}}(f_i) - \Phi_{s_{\text{eff},i}}(f_{i+1})}{\Phi_{s_{\text{eff},i}}(f_i)}, 0\right) \quad (4)$$

where f_i and f_{i+1} denote estimated SDF values at the entry and exit of the i-th interval, computed using the cosine-annealed distance approximation from NeuS. The key distinction from prior work is that $s_{\text{eff},i}$ is computed per-

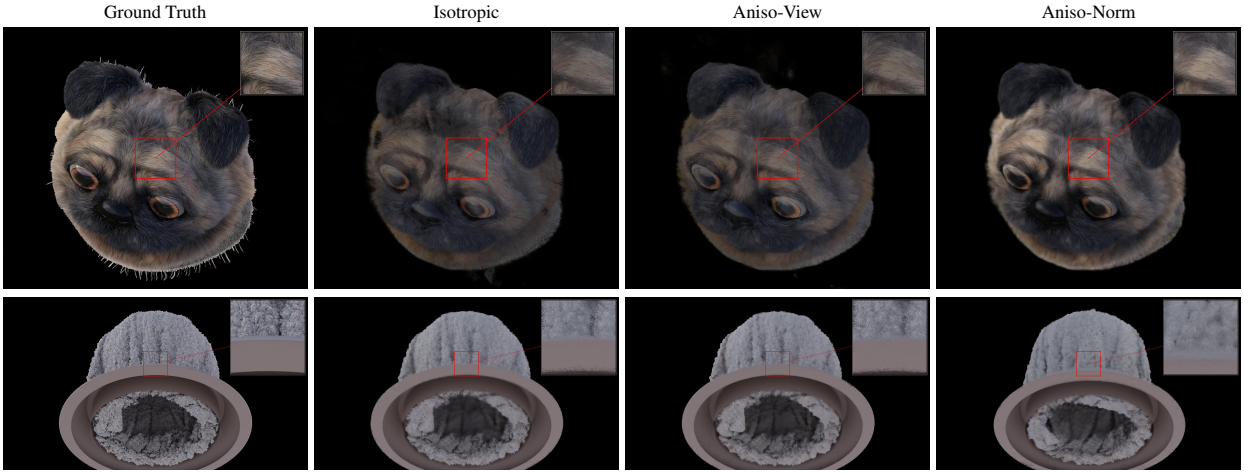

Figure 1. Qualitative results on the test-views of *Shelly* dataset, specifically for *Pug* and *Woolly*

interval using the anisotropic kernel and directional projection.

For hierarchical importance sampling, we similarly compute interval-wise effective kernels during upsampling. When using normal projection, we evaluate surface normals ∇f at interval midpoints through finite differences of the SDF field, which are then used to compute s_{eff} via Eq. 3.

3.5. Training

Network Architecture Following Adaptive Shells [11], our model represents the scene using a neural field composed of a multi-resolution hash encoding [8] with shallow MLP decoders. Given a 3D location $\mathbf{x}$ and viewing direction $\mathbf{d}$, the network predicts the SDF value f, view-dependent color $\mathbf{c}$, and auxiliary outputs such as geometry features and predicted surface normals. Our architecture and training setup are identical to [11] unless stated otherwise. Critically, the only architectural modification we introduce is the kernel projection used for the density function. We extend the kernel size output dimension from a scalar $s \in \mathbb{R}$ (isotropic) to a 3D vector $\mathbf{s} \in \mathbb{R}^3$ (anisotropic), enabling directionally-adaptive density spread around the surface.

Loss Functions Our optimization follows the loss formulation introduced in Adaptive Shells [11]:

$$\mathcal{L} = \mathcal{L}_{\text{color}} + \lambda_e \mathcal{L}_{\text{eikonal}} + \lambda_s \mathcal{L}_{\text{smooth}} + \lambda_n \mathcal{L}_{\text{normal}} \quad (5)$$

Here, $\mathcal{L}_{\text{color}}$ denotes the pixel-wise reconstruction loss measured using the L1 distance. The eikonal loss $\mathcal{L}_{\text{eikonal}}$ enforces that the learned function f behaves as a valid signed distance field by regularizing the gradient norm toward 1. The smoothness term $\mathcal{L}_{\text{smooth}}$ penalizes rapid spatial variation in the kernel field and is applied independently to each component of $\mathbf{s}$. Finally, the normal alignment loss $\mathcal{L}_{\text{normal}}$ encourages consistency between predicted normals and SDF gradients. Importantly, we omit the erosion–dilation shell extraction step used in [11], as our goal is to isolate and study the effect of the kernel representation itself rather than perform rendering acceleration.

4. Experiments

In this section, we present implementation details and assess the rendering quality of both approaches on the *Shelly* dataset. As a baseline, we also compare against an isotropic learned density kernel trained with identical hyperparameters.

4.1. Implementation Details

Following [9], we bound the region of interest inside a unit sphere and train a separate NeRF [7] model for the background. We sample 4096 rays per batch and train for $200k$ iterations. We also use AdamW optimizer with weight decay of 1×10^{-2}. The learning starts at 1×10^{-3}, is linearly warmed up to 5×10^{-3} over the first 5000 iterations, and decays exponentially to 1×10^{-4}. We adopt the loss terms from NeuS [9] and Adaptive Shells [11] with the same weighting as Adaptive Shells [11]: $\lambda_e = 0.01, \lambda_s = 0.01$ and $\lambda_n = 0.1$.

4.2. Evaluation Metrics

We assess visual quality using the three standard metrics: the peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) and learned perceptual image patch similarity (LPIPS). Our evaluation is conducted on the synthetic *Shelly* dataset introduced in [11], focusing on the *Pug* and *Woolly* object-level scenes. These scenes contain fuzzy surfaces (fur, wool) as well as smooth areas, making them suitable for evaluation.

Table 1. Quantitative results on the *Shelly* dataset. We report the PSNR, LPIPS and SSIM results for two scenes and compare across different kernels; we highlight the best results.

	Pug			Woolly		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Isotropic	21.63	0.736	0.247	25.94	0.815	0.142
View Direction Projection	21.60	0.729	0.263	25.79	0.822	0.139
Normal Direction Projection	24.20	0.832	0.178	25.60	0.812	0.149

5. Discussion

Table 1 compares isotropic and directional kernel parameterizations on two scenes from the *Shelly* dataset. In the *Pug* scene, the Normal Direction Projection achieves the highest performance across all quantitative metrics. The *Pug*'s fur exhibits a consistent strand direction over small spatial neighborhoods, producing stable normals within each kernel's support. This makes normal-aligned anisotropy well-defined and effective for capturing the elongated geometry of the fur. Aligning the anisotropic kernel with the SDF gradient provides a meaningful signal that captures the elongated structure of the fur more effectively than an isotropic or view-aligned kernel.

In contrast, the *Woolly* scene's geometry is dominated by dense, fuzzy microstructure with highly irregular and rapidly varying local normals. Because this structure is largely anisotropic and lacks clear orientation cues, normal-aligned anisotropy provides a relatively weak signal for training the extra parameters. The appearance of such fibrous surfaces is strongly-view dependent due to self-occlusion, scattering, and grazing-angle effects. As a result, the View Direction Projection is more effective at capturing perceptual detail, explaining why it outperforms the Normal Direction Projection in SSIM and LPIPS on this scene.

5.1. Limitations

Our current formulation has several limitations. First, adopting an anisotropic kernel increases the parameter dimensionality, from a single scalar to a 3D vector, thereby enlarging the optimization space and introducing additional challenges for stable and efficient convergence. Second, because our evaluation spans only two scenes, it is difficult to draw definitive conclusions about the broader conditions under which anisotropy is beneficial. A more diverse set of geometries would help validate the generality of our observations.

6. Conclusion

In this work, we investigated anisotropic kernel representations for neural implicit surface reconstruction, extending spatially-varying scalar kernels to directionally-adaptive vector formulations. Our evaluation demonstrates that incorporating directional information into SDF-to-density conversion can substantially improve reconstruction quality when geometric structure aligns with the projection strategy. Normal-aligned anisotropic kernels achieve significant improvements (2.57 dB PSNR) on directionally-structured fur, while isotropic representations remain competitive for highly irregular volumetric regions.

The choice of projection strategy fundamentally affects performance: normal projection excels for coherent directional structure, while view projection may better capture appearance variations in highly anisotropic materials. Future work could integrate anisotropic kernels into adaptive shell extraction for efficient narrow-band sampling. While our study is exploratory and limited in scope, the substantial improvements on strand-like features establish that directional kernel adaptation merits continued investigation for modeling fuzzy geometry in neural implicit surfaces.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021. 1

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. 1

[3] Stefano Esposito, Anpei Chen, Christian Reiser, Samuel Rota Bulò, Lorenzo Porzi, Katja Schwarz, Christian Richardt, Michael Zollhoefer, Peter Kontschieder, and Andreas Geiger. Volumetric surfaces: Representing fuzzy geometries with layered meshes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2

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

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 2

[6] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[7] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 3

[8] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 1, 2, 3

[9] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2021. Curran Associates Inc. 1, 2, 3

[10] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1

[11] Zian Wang, Tianchang Shen, Merlin Nimier-David, Nicholas Sharp, Jun Gao, Alexander Keller, Sanja Fidler, Thomas Müller, and Zan Gojcic. Adaptive shells for efficient neural radiance field rendering. *ACM Trans. Graph.*, 42(6), 2023. 1, 2, 3

[12] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 1