
NEURAL DENSITY-DISTANCE FIELDS

Itsuki Ueda

University of Tsukuba

ueda.itsuki@image.iit.tsukuba.ac.jp

Yoshihiro Fukuhara

Waseda University

f_yoshi@ruri.waseda.jp

Hirokatsu Kataoka

National Institute of Advanced Industrial Science and Technology

hirokatsu.kataoka@aist.go.jp

Hiroaki Aizawa

Hiroshima University

hiroaki-aizawa@hiroshima-u.ac.jp

Hidehiko Shishido

University of Tsukuba

shishido.hidehiko@image.iit.tsukuba.ac.jp

Itaru Kitahara

University of Tsukuba

kitahara.itaru@image.iit.tsukuba.ac.jp

ABSTRACT

The success of neural fields for 3D vision tasks is now indisputable. Following this trend, several methods aiming for visual localization (e.g., SLAM) have been proposed to estimate distance or density fields using neural fields. However, it is difficult to achieve high localization performance by only density fields-based methods such as Neural Radiance Field (NeRF) since they do not provide density gradient in most empty regions. On the other hand, distance field-based methods such as Neural Implicit Surface (NeuS) have limitations in objects' surface shapes. This paper proposes Neural Density-Distance Field (NeDDF), a novel 3D representation that reciprocally constrains the distance and density fields. We extend distance field formulation to shapes with no explicit boundary surface, such as fur or smoke, which enable explicit conversion from distance field to density field. Consistent distance and density fields realized by explicit conversion enable both robustness to initial values and high-quality registration. Furthermore, the consistency between fields allows fast convergence from sparse point clouds. Experiments show that NeDDF can achieve high localization performance while providing comparable results to NeRF on novel view synthesis. The code is available at <https://github.com/ueda0319/neddf>.

1 Introduction

Representing 3D shapes using coordinate-based neural networks, also known as neural fields [26] have recently attracted attention as an alternative to using point clouds, voxels, meshes, and others [14, 20, 22, 19, 18, 3, 24]. Neural Radiance Fields (NeRF) [14], in particular, have shown impressive quality for tasks such as novel view-synthesis. However, since NeRF has limited regions with smooth spatial density and color, many conventional methods still require good initial values for registration and localization tasks. This paper proposes a distance field representation that is reciprocally constrained to the density field, named Neural Density-Distance Field (NeDDF). NeDDF achieves robust localization with distance fields while providing object reconstruction quality comparable to NeRF.

As shown in Fig. 1, there are two main types of 3D shape representation in neural fields: density field used in NeRF [14] and distance field used in NeuS [24]. Density field has high expressiveness for translucent objects, such as smoke and water, and high-frequency shapes, such as hair. However, in most areas except the boundary, the gradient of the field is zero. This makes it difficult to set up a convex objective function in a problem setting such as registration, as shown in Fig. 2. Distance field provides the gradient over a wide range even after the optimization converges. Thus, we can establish objective functions with high convexity in registration. The field can be learned from the image via volume rendering by defining a conversion equation from distance to density. For example, NeuS assumes that the density

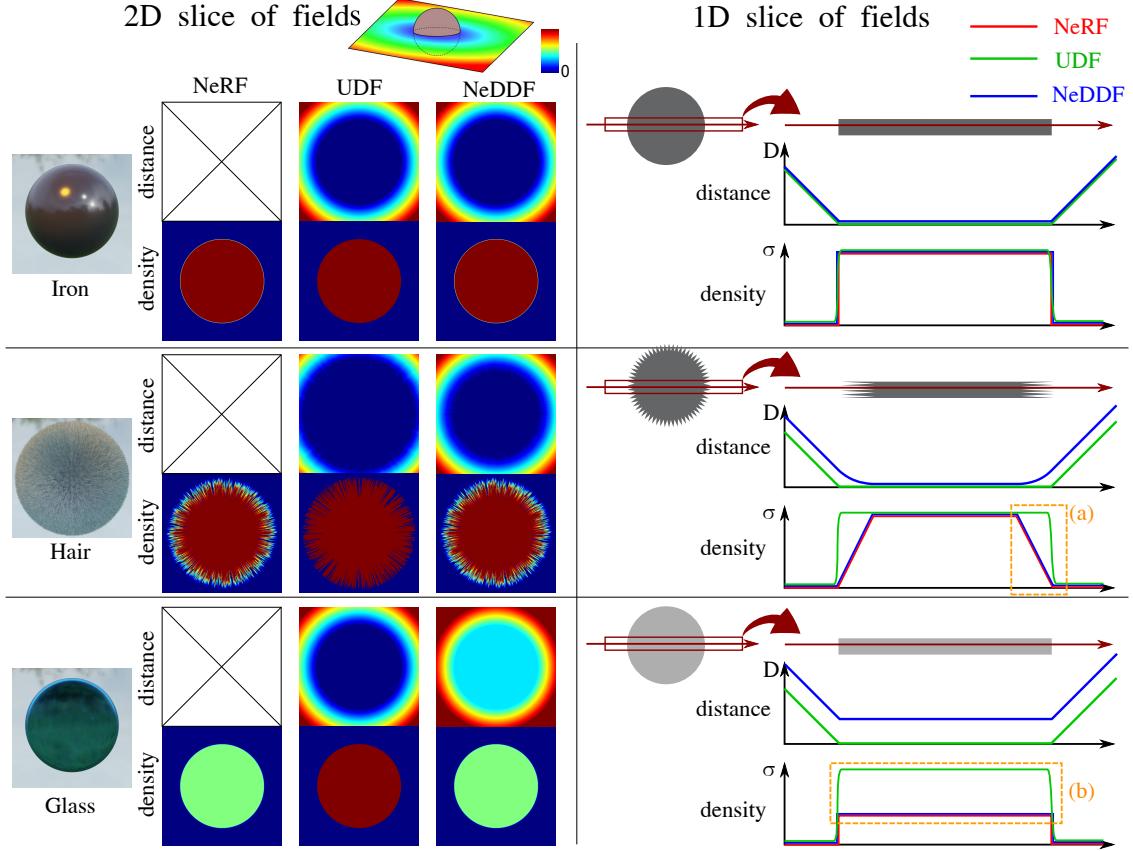


Figure 1: (left) Visualization of the 2D slice for each field with iron, hair, and glass spheres as examples. (right) Plots of 1D slices for each field. NeRF provides no distance information. Unsigned Distance Field (UDF) cannot handle some cases correctly, such as (a) ambiguous density changes such as a hairball or (b) low densities such as a glass ball. Proposed NeDDF can represent both cases properly.

follows a logistic distribution close to the object surface. On the other hand, since we assume explicit boundaries, the convertible density field is tightly restricted.

As shown in Fig. 3 (a), we focus on the Unsigned Distance Field (UDF), which ignores surface direction inside objects and can distinguish between the inside and outside objects not only by the sign of distance D but also by the magnitude of its gradient. We extend the distance field to be able to recover arbitrary density distributions by interpreting D by the depth derived from the volume rendering equation and fitting the density information of translucent objects to the mid-level gradient magnitude, as shown in Fig. 3 (b). This method eliminates the need for constraints on the density, as in NeuS, when learning the distance field from images. In other words, when learning the density field, we can simultaneously obtain a consistent distance field where the shape and camera pose have the same optimal values. As shown in Fig. 4, the NeDDF has a network that inputs a position and outputs the distance and its gradient, and a converter that explicitly calculates the density. NeDDF enables both high expressiveness of the density field and good registration of the associated distance field.

The present paper provides the following three contributions: (1) Extending the distance field to be definable for arbitrary density distributions. (2) Proposing a method to recover the corresponding density from independent points using the distance field and gradient information. (3) Providing an implementation to alleviate the instability of the distance gradient caused by cusp points and sampling frequency. Furthermore, the effectiveness of the proposed method in terms of both expressiveness and registration performance is evaluated through experiments.

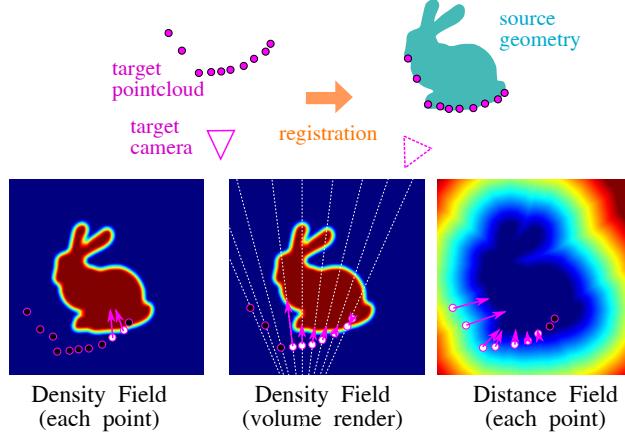


Figure 2: Registration following in each representation: white and black points (small circles) denote the points where the gradient direction is available or not, respectively. In the density field (left), point-by-point gradients cannot be obtained in most regions except for the boundary. In volume rendering of the density field (center), it is impossible to obtain the gradient component in the vertical direction of view. In the distance field (right), we can obtain both the gradient direction and the residuals for each point.

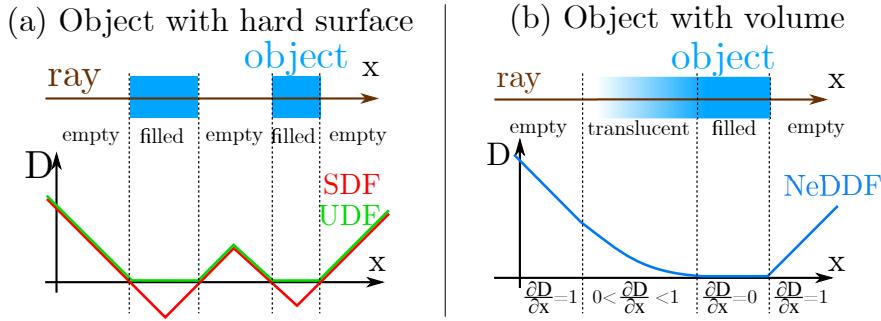


Figure 3: (a) Signed Distance Field (SDF) and Unsigned Distance Field (UDF) distinguish the inside and outside of objects by the sign of the distance field ($D > 0$ or $D = 0$). In addition, UDF does not have a gradient inside the object, but can distinguish between inside and outside of objects by the magnitude of the gradient. (b) NeDDF assigns semitransparent density information to the distance gradient from 0 to 1.

2 Related Work

2.1 Neural Fields

The traditional way of representing volumes is to discretize the density or distance from the surface at each position into voxels [16, 15, 10, 9, 8, 5]. Since voxels require data complexity that is cubic in resolution, it is difficult to increase the spatial resolution. In recent years, memory-efficient representations such as octree [2, 7] or hash table [17] have been proposed. However, these grid structures cannot represent the geometry information with higher frequencies than the Nyquist frequency. With advances in geometric deep learning following AtlasNet [6] and Foldingnet [27], some studies have focused on handling irregular non-grid structures such as point clouds and meshes. While these methods can handle detailed geometry information efficiently, they are limited to interpolation due to their spatially discrete representations based on the coordinates of each point and vertex.

In recent years, the methods called neural fields [26] which directly represent continuous signals by embedding implicit surfaces with neural networks, has been attracting significant interest in the research community. Given sufficient parameters, fully connected neural networks can encode continuous signals over arbitrary dimensions. Since a neural field automatically adapts the expressiveness of the network to high frequency regions, it is possible to obtain a shape representation with high resolution using a significantly smaller number of parameters than conventional discrete representation based methods. Furthermore, since neural fields are continuous representation, we can expand the input

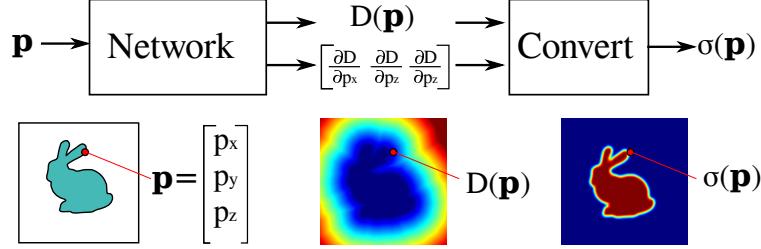


Figure 4: Flow of proposed model: created network takes the position \mathbf{p} as input and outputs the distance D and its gradient. Differentiable conversion formula for the network output gives the density of position $\sigma(\mathbf{p})$.

and output dimensions without increasing the model capacity. It is also possible to model topological changes by considering the density field as a three-dimensional slice of high dimensional space and embed time-series information by adding a multi-dimensional deformation code to the input [21].

By using a smooth activation function in the neural networks, neural fields can be regarded as a continuously differentiable field. Modeling using gradient information has been proposed, such as divergence in Non-Rigid-NeRF [23] and elasticity constraints using the Jacobian matrix of the deformation field in Nerfies [20]. Inspired by the ideas behind these approaches, we have developed a model in which the gradient of the distance field describes the density information. The proposed model outputs a distance field and a density field that are explicitly consistent with each other. Since an optimization-based penalty term does not constrain the model, it can be optimized reciprocally from an objective function that is appropriate for each field.

2.2 Density Field

The density field outputs the volume density for the input of the 3D position. Many methods, such as NeRF, use the density field together with the color field, thus enabling volume rendering. The density field is characterized by high expressiveness. For example, a low value of the density field can describe a semi-transparent object such as glass or smoke to represent proportional light transmission. For spatially high-frequency shapes such as fur, for which the boundary surface is complex, the field can describe a scene by considering the light interaction at an arbitrary point as a function of density, ray direction, and color. In particular, combining this with the color field makes it possible to model specular reflections, including viewing angle dependency.

Although NeRF can describe complex scenes, it has a substantial limitation in that the camera pose should be known for the observed image and the scene should be static, making it challenging to capture a usable set of images with unknown camera poses. Therefore, many NeRF-based methods for estimation of camera pose and registration of object deformation have been proposed [30, 12]. However, since blank areas with a density value of 0 have uncertain gradient directions, camera pose tracking is only valid with initial values such that most of the object area overlaps. NeRF— [25] optimizes camera parameters directly with backpropagation but is limited to camera configurations close to the line of sight. NSFF [11] requires the optical flow to follow the deformation. D-NeRF is limited to CG images with no background and low-frequency texture. Nerfies [20] adds warmup to positional embedding and delays learning of high-frequency components to ensure registration but is limited to camera configurations with close view directions.

NeDDF provides a consistent distance field while retaining the expressiveness of the density field. By providing gradients where no objects are present, we can improve registration performance from a rough initial camera poses.

2.3 Distance Field

A distance field takes a 3D position as input and outputs the distance to the nearest neighbor boundary. SDF is widely used in fusion and registration because it can provide stable bounding surfaces and normal vectors. SDF also provides residuals and gradient directions, enabling fast-fitting of two shapes by the Gauss-Newton method without corresponding point matching [16, 15, 10, 9, 8, 5]. A typical example is KinectFusion [16], which performs localization and shape integration from a depth map of unknown viewpoints for a static scene. DynamicFusion [15] constructs a sparse deformation field called WarpField to describe the deformation amount and performs registration for non-rigid scenes.

In addition, several studies have proposed methods by which to handle distance fields by neural fields. DeepSDF [19] proposes a generative model for the continuous SDFs based on the auto-decoder model. SAL [1] enables the neural

field to learn the shapes of boundary surfaces directly from raw unsigned data such as point clouds. UDF [3] makes the unsigned distance field continuous and shows its suitability for unclosed surfaces and complex shapes. Since density fields such as NeRF are noisy in surface reconstruction by level sets, several methods have been proposed to handle distance fields in neural fields that can present boundary surfaces and assume a static density distribution for the signed distance. IDR [29] introduces differentiable surface rendering to learn the neural field from multi-view images. However, unlike volume rendering, calculating a single surface intersection for each ray makes shape reconstruction unstable for complex shapes that cause abrupt depth changes in the image. UNISURF [18] enables a combination of surface and volume rendering with a neural field that describes occupancy. Several studies attempt to optimize the SDF with differentiable volume rendering by modeling the transform equation from the distance to the density field with the hypothesis on the shape of the density distribution. VolSDF [28] interprets the volume density as Laplace's cumulative distribution function for SDF. NeuS [24] assumes a density distribution over the signed distance using the learnable variance values. However, these distance fields assume the existence of a boundary surface, which limits the kind of shape that can be represented. This study extends the distance field to correspond to various density distributions from depth values derived from volume rendering. The present extends the distance field to correspond to various density distributions from depth values derived from volume rendering.

3 Method

In this section, we consider reciprocally constrained distance and density fields. Section 3.1 redefines the distance field to interpret arbitrary density fields, including boundaryless scenes. Section 3.2 introduces a conversion formula to obtain the density of independent points from the distance and the gradient of the distance value in the redefined distance field.

3.1 Distance Field from Density Field

The distance field in boundary surfaces $D_b(\mathbf{p})$ describes the distance to the nearest surface for location $\mathbf{p} \in \mathbb{R}^3$. We can interpret $D_b(\mathbf{p})$ as the minimum of the depth value $d_b(\mathbf{p}, \mathbf{v})$ over the viewing direction $\mathbf{v} \in \mathcal{S}^2$ (see Fig. 5):

$$D_b(\mathbf{p}) := \min_{\mathbf{v} \in \mathcal{S}^2} [d_b(\mathbf{p}, \mathbf{v})]. \quad (1)$$

We extend the distance field to be defined for arbitrary density distributions by replacing the depth value $d_b(\mathbf{p}, \mathbf{v})$ with the depth value derived from the volume rendering equation (see Fig. 6).

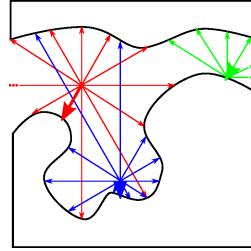


Figure 5: The distance field uses depth for the nearest surface.

Assuming that the density distribution $\sigma(\mathbf{p})$ is known, we can calculate the depth value $d(\mathbf{p}, \mathbf{v})$ for a viewpoint position \mathbf{p} and a viewing direction \mathbf{v} . Equation 2 is the volume rendering formula in NeRF [14]. For a point on the ray $\mathbf{r}(t) = \mathbf{p} + t\mathbf{v}$ with the visible range $[t_n, t_f]$, the color of each ray $\mathbf{C}(\mathbf{r})$ is obtained through the integral of each color $\mathbf{c}(\mathbf{r}(t), \mathbf{v})$ multiplied by transmission rate $T(t) = \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}(s))ds\right)$:

$$\mathbf{C}(\mathbf{r}) := \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{v})dt. \quad (2)$$

Similarly, the depth $d(\mathbf{p}, \mathbf{v})$ is defined to be an integral of the depths at each point, as follows:

$$d(\mathbf{p}, \mathbf{v}) := t_n + \int_{t_n}^{t_f} tT(t)\sigma(\mathbf{r}(t))dt. \quad (3)$$

Here, $d(\mathbf{p})$ takes the same value for the depth value $d_b(\mathbf{p})$ in the presence of a boundary surface by taking the density $\sigma(\mathbf{p})$ to be 0 outside ($0 < t < d_b$) and ∞ inside ($d_b \leq t$).

In practice, calculating the depths over all directions is computationally expensive, so we use a distance field that removes the dependence on the viewing direction \mathbf{v} . We define the distance field $D(\mathbf{p}) := \min_{\mathbf{v} \in S^2} [d(\mathbf{p}, \mathbf{v})]$, for adopting the shortest depth for each position as in the bounding surface model. Assuming continuity in the adopted view direction $\mathbf{v}_n = \operatorname{argmin}_{\mathbf{v} \in S^2} [d(\mathbf{p}, \mathbf{v})]$, we can restore this quantity from the tangent plane using the gradient of the distance field $\nabla D(\mathbf{p})$ as follows:

$$\mathbf{v}_n = \frac{-\nabla D(\mathbf{p})}{\|\nabla D(\mathbf{p})\|_2}, \quad (4)$$

$$\nabla D(\mathbf{p}) = \left[\frac{\partial D(\mathbf{p})}{\partial p_x} \quad \frac{\partial D(\mathbf{p})}{\partial p_y} \quad \frac{\partial D(\mathbf{p})}{\partial p_z} \right]. \quad (5)$$

In practice, \mathbf{v}_n has discontinuities which makes this calculation difficult. In the next section, we will discuss some strategies to alleviate this.

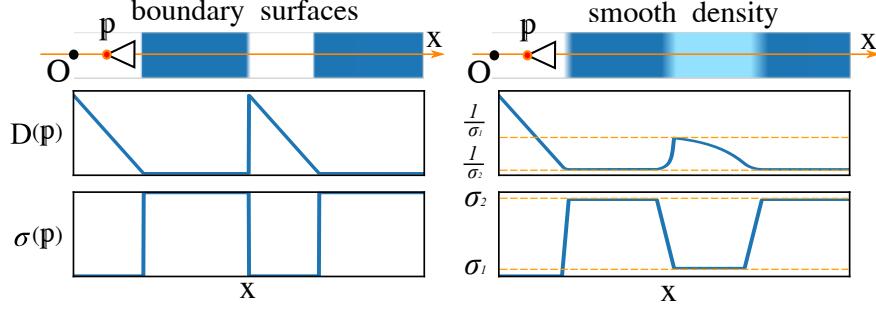


Figure 6: Depth of volume rendering in case camera is on the x-axis. In a translucent region of density σ , the upper bound of the depth value becomes $1/\sigma$. When σ takes ∞ inside the object, the depth value takes 0, corresponding to the conventional hard surface model.

3.2 Density from Distance Field

In Section 3.1, we extended the distance field to shapes with no explicit boundary surface but with a known density field. In this section, we derive the corresponding density field when the distance field is known.

For the distance field around position \mathbf{p} , we consider $D(\mathbf{r}(t))$, $\mathbf{r}(t) = \mathbf{p} + t\mathbf{v}$, which is sliced in the gradient direction \mathbf{v} . Calculating the derivative of the distance field in the direction of the gradient, $\frac{\partial D(\mathbf{r}(t))}{\partial t}$, we can derive an expression for σ as follows: (the derivation is given in supplementary material)

$$\frac{\partial D(\mathbf{r}(t))}{\partial t}|_{t=0} = \lim_{\Delta t \rightarrow 0} \frac{d(\mathbf{r}(\Delta t), \mathbf{v}) - d(\mathbf{r}(0), \mathbf{v})}{\Delta t} \quad (6)$$

$$= -1 + (D(\mathbf{p}) - t_n)\sigma(\mathbf{p} + t_n\mathbf{v}). \quad (7)$$

We can also express $\frac{\partial D(\mathbf{r}(t))}{\partial t}$ using the gradient vector of the distance field $\nabla D(\mathbf{p})$ as follows:

$$\frac{\partial D(\mathbf{r}(t))}{\partial t} = \frac{\partial D}{\partial p_x} \frac{\partial p_x}{\partial t} + \frac{\partial D}{\partial p_y} \frac{\partial p_y}{\partial t} + \frac{\partial D}{\partial p_z} \frac{\partial p_z}{\partial t} \quad (8)$$

$$= \nabla D(\mathbf{p}) \cdot \mathbf{v} \quad (9)$$

$$= -\|\nabla D(\mathbf{p})\|_2. \quad (10)$$

From Equations 7 and 10, the density σ can be obtained as follows:

$$\sigma(\mathbf{p}) = \frac{1 - \|\nabla D(\mathbf{p} - t_n\mathbf{v})\|_2}{D(\mathbf{p} - t_n\mathbf{v}) - t_n}. \quad (11)$$

From Equation 3, t_n is the interval in which the transmittance $T(t_n) = 1$, which is the lower limit of D . While it is natural for t_n to take the value 0, the density is undefined for $D \sim 0$ as a numerical problem. Assuming t_n is sufficiently small, by introducing the approximations in Equations 12,13, σ can be calculated as in Equation 14:

$$D(\mathbf{p} - t_n\mathbf{v}) \simeq D(\mathbf{p}) + t_n, \quad (12)$$

$$\nabla D(\mathbf{p} - t_n\mathbf{v}) \simeq \nabla D(\mathbf{p}), \quad (13)$$

$$\sigma(\mathbf{p}) \simeq \frac{1 - \|\nabla D(\mathbf{p})\|_2}{D(\mathbf{p})}. \quad (14)$$

The differentiability of the network allows us to determine the distance $D(\mathbf{p}_i)$ and gradient vector $\nabla D(\mathbf{p}_i)$ with independent sampling points \mathbf{p}_i as input to the neural fields, such as regressing a distance field. Equation 14 allows us to compute the density $\sigma(\mathbf{p}_i)$ with a differentiable conversion formula. In other words, it is possible to learn mutually constrained distance and density fields. Note that from $t_n > 0$, the density is limited to $\sigma(\mathbf{p}) < \frac{1}{t_n}$.

3.3 Removing Cusps

The distance to density conversion by Equation 14 assumed that the distance field is first-order differentiable. However, a distance field is not differentiable at the cusps where the minimum distance direction switches. In practice, although the distance field around the cusps is smoothly connected due to the continuity of the neural field, small gradient values around the cusps are still converted to false densities that should not exist by Equation 14 (see Fig. 7). To alleviate this problem at the caps, we extend the domain of the distance field from a 3-dimensional space to a 4-dimensional hyperspace $[x, y, z, w]$ with an auxiliary gradient axis (w -axis) and consider the slice at $w = 0$ of the hyperspace to be the distance field. The gradient components $\frac{\partial D}{\partial w}$ are distributed so that the gradient ∇D satisfies the Equation 14 in the vicinity of the cusp to suppress spurious densities.

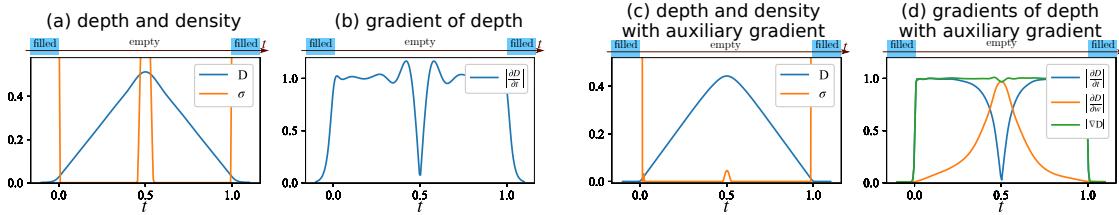


Figure 7: A visualization of each field in the toy model with $0 \leq t \leq 1$ as object exterior.

When the auxiliary gradient also describes the density outside the cusp region, the shape of the global distance field is compromised. Therefore, we constrain the shape of the auxiliary gradient and use a penalty term to induce convergence to the optimal form. In this study, when $\frac{\partial D}{\partial t} > 0$, we use the following heuristic constraint:

$$\frac{\partial^2 D}{\partial t \partial w} = \alpha \frac{1}{D} \frac{\partial D}{\partial w}. \quad (15)$$

Note that α is a hyperparameter that determines the scale of the gradient. The shape of the auxiliary gradient for each α is shown in appendix. Since the Equation 15 is a constraint other than $\frac{\partial D}{\partial t} = 0$, and since it becomes unstable around $D = 0$ and $\frac{\partial D}{\partial w} = 0$, we introduce the following weight coefficient β :

$$\beta = D \left(\frac{\partial D}{\partial t} \right)^2 \frac{\partial D}{\partial w}. \quad (16)$$

Note that since β is a value for discriminating the target point, it is separated from the calculation graph during training like stop-gradient operator. For M sampling points \mathcal{P} , the objective function L_{const} for the shape constraint of the auxiliary gradient is set as:

$$L_{\text{const}} = \frac{\lambda_{\text{const}}}{M} \sum_{\mathbf{p} \in \mathcal{P}} \beta \left[\frac{\partial^2 D}{\partial t \partial w} - \frac{\alpha}{D(\mathbf{p})} \frac{\partial D}{\partial w} \right]^2. \quad (17)$$

Note that λ_{const} is a hyperparameter.

4 Reprojection error for volume rendering

Previous NeRF-based localization such as iNeRF [30] uses photometric error, the residual $\|\mathbf{C}(\mathbf{q}) - \hat{\mathbf{C}}(\mathbf{q})\|_2$ from the observed color $\mathbf{C}(\mathbf{q})$, and the estimated color $\hat{\mathbf{C}}(\mathbf{q})$ aggregated by volume rendering for selected pixel \mathbf{q} . While photometric error can follow objects without hard surfaces, it can only follow local regions with smooth color changes. NeDDF provides the object’s direction and approximate distance from a sampling point. Therefore, we can calculate the pseudo-correspondence point and estimate the camera pose using the reprojection error. This section describes a method that uses color information to calculate the correspondence points as a simple example.

When training the network, the objective function with volume rendering has no constraints about the color field in the blank region. We record the same color in the gradient direction of the distance field by penalizing the color change in

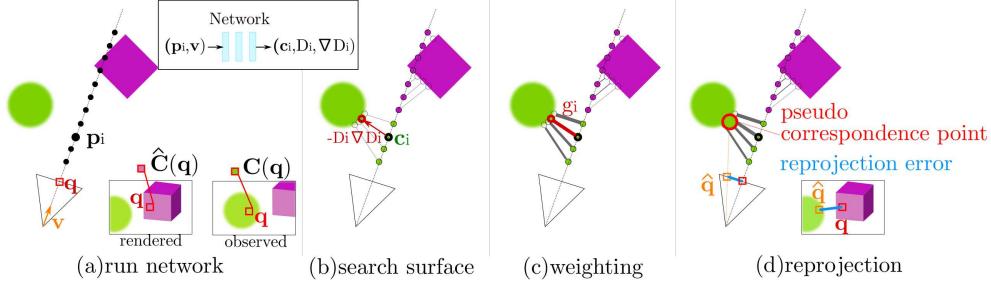


Figure 8: Localization by reprojection error: (a) Network execution outputs the color and distance of each sampling point. Network provides the color c_i , distance D_i and its gradient ∇D_i for the sampling point p_i . (b) Network output gives the points near the surface. (c) Weight g_i takes the value that emphasizes sampling points with similar colors or smaller distances. (d) Pseudo-correspondence points synthesized from the weights produce the reprojection error.

the distance gradient direction for a blank region. For a point p_i , camera depth t_i and a viewing direction v , let the output be color c_i , distance D_i , and distance gradient ∇D_i . The penalty takes $L_{\text{blank}} = \sum_i \|\nabla c_i(\nabla D_i)^T\|_2$. Since $\|\nabla D_i\|_2$ takes small values inside the object, this penalty becomes active for regions outside the object. Training the network by introducing this penalty makes it possible to obtain the nearest neighbor object’s color, direction, and distance from sampling points outside the object.

In estimating camera poses, finding the pseudo-correspondence points for each ray provides the calculation of the reprojection error, as shown in the Fig. 8. The network output for sampling point p_i leads to the near-surface point of color c_i at position $p_i - D_i(\nabla D_i)$. We calculate the pseudo-correspondence points for each ray by selecting points closer to $C(q)$ than $\hat{C}(q)$ with combining near-surface points by focusing weights closer to the color and the distance between them. The weight g_i of sampling point p_i is calculated as follows:

$$g_i = \text{softmax}\left(-\lambda_D \frac{D_i \|\nabla D_i \times v\|_2}{t_i} - \lambda_c \|C(q) - c_i\|_2\right). \quad (18)$$

Note that λ_D and λ_c are hyperparameters, softmax aggregates for axis with index i. The reprojection error measures the distance $\|\mathbf{q} - \hat{\mathbf{q}}\|_2$ between the pixel coordinates of the ray q and the projected pseudo-correspondence point \hat{q} .

5 Experiments

5.1 Experimental setup

5.1.1 (a) Quality of novel view synthesis.

We confirm that NeDDF retains the comparable quality of novel view synthesis as NeRF. We use the NeRF synthetic dataset which contains CG images of the eight scenes rendered from camera positions placed on a hemispherical surface. Using NeRF and NeuS as baselines, we compare the quality of the new viewpoint images using PSNR.

5.1.2 (b) Accuracy of localization.

In NeDDF, the pseudo correspondence points enable estimation of camera pose using reprojection error. We confirm that the use of reprojection error improves the accuracy of localization from only using photometric error under poor initial camera poses. The experiment uses 200 test viewpoints of the Lego scene in the NeRF synthetic Dataset. Each camera takes the initial pose given by random rotations and translations applied to the ground truth values of the test viewpoint. We evaluate the camera position and angle errors for three cases: optimizing the camera pose for 300 iterations by photometric error, 300 iterations by reprojection error, and 100 iterations by reprojection error plus 200 iterations by photometric error. The localization flows follow similar practices to iNeRF. The optimization uses the gradient of the camera pose as a 6D parameter of SE(3) and sets the increment by Adam [4] with exponential decay. In each iterations, interest region sampling [30] selects 256 rays for optimization.

Table 1: Quantitative evaluation on synthetic dataset. We report PSNR (higher is better).

Method	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Mean
NeRF	33.00	25.01	30.13	36.18	32.54	29.62	32.91	28.65	31.01
NeuS	27.69	22.14	21.67	32.14	27.18	25.64	27.52	23.47	25.93
NeDDF	29.11	23.96	25.72	30.85	27.93	25.52	29.34	23.69	27.02

5.2 Results

5.2.1 (a) Quality of novel view synthesis.

Table 1 shows the PSNR for each scene and each method for evaluating the quality of the generated images. The NeDDF results retain the comparable quality of the state-of-the-art methods in novel view synthesis, although the PSNR comparison is slightly inferior. Fig. 9 shows visualizations of Drums and Ficus as examples of scenes with transparency and delicate shapes, where conventional distance-field-based methods are weak. For the Drum scene, NeuS could not represent the transparent parts of the drums and colored the transparent surfaces. In contrast, NeDDF reproduces the transparent parts and obtains the metal parts on the back through. The NeuS results show blurred images for the Ficus scene since it fails to assign the appropriate density distributions due to the difficulty of understanding the model for thin surfaces such as leaves. NeDDF provides comparatively high-quality image restoration even for delicate shapes such as leaves. NeDDF allows for the reconstruction of thinner surfaces than the sampling interval by interpreting their occupancy as a lower density.

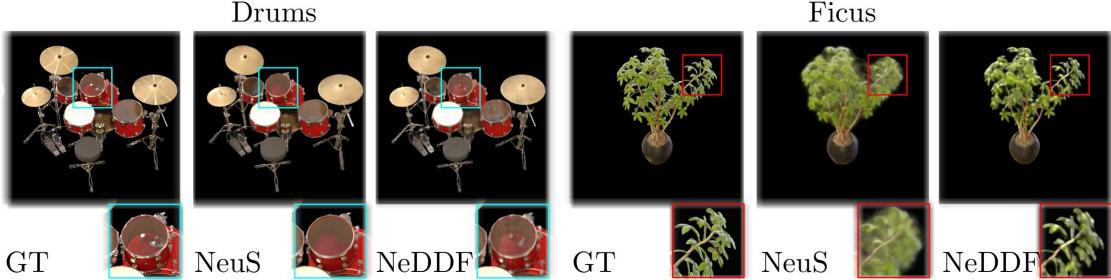


Figure 9: Comparisons on test-set views for scenes from NeRF synthetic dataset generated with a volume renderer.

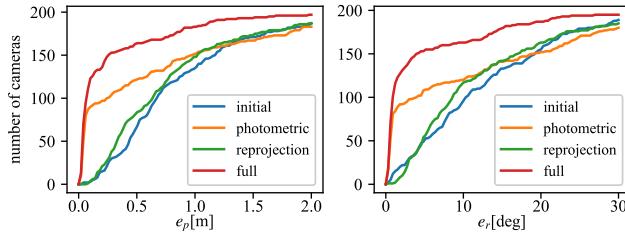


Figure 10: Quantitative evaluation of camera poses estimation accuracy. The horizontal axis represents the position and angle error, and the vertical axis represents the number of cameras recovered under the errors.

5.2.2 (b) Accuracy of localization.

Fig. 10 shows a plot of the camera attitude estimation results. Optimization by photometric error works very accurately when the initial value of the position error is approximately 0.5 [m] or less, and the initial value of the angular error is 10 [deg] or less, while the error increases when the initial value of the error is significant. Since photometric error only provides reasonable gradient directions in smooth color changing, significant initial value errors may lead to erroneous local solutions. Optimization by reprojection error does not improve camera posture residuals much on its own. This is because the correspondence point calculation based on color similarity lacks uniqueness, which leads to many mismatches in the correspondence points. On the other hand, optimization by reprojection error can roughly

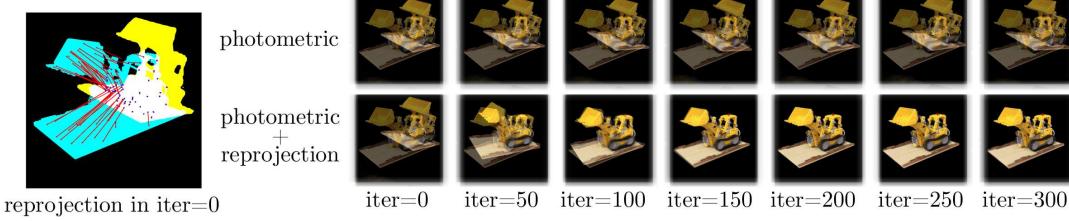


Figure 11: (left) Projection of the corresponding points in iteration 0: the cyan region is the rendered object region of the observed image, and the yellow region is the rendered object region estimated from the current camera attitude. The red dots denote sampling points, and the blue dots denote the projected positions of the calculated correspondence points. (right) Observed and rendered images at each optimization iteration: the upper row shows the case optimized using only photometric error, and the lower row shows the case optimized using the photometric and reprojection errors.

estimate the camera pose such that regions with close colors overlap. Using this as a preprocessing step for optimization by photometric error can significantly expand the range of initial values for highly accurate estimation.

Fig. 11 shows the localization process with large initial value errors. The left figure shows the projected positions of the sampling points and the calculated correspondence points at iter=0. The figure shows that even when the sampling points are far from the object, as in the lower-left region, we can obtain the corresponding points in the object region. On the other hand, in the upper left area of the bucket, the model selects the wrong correspondence points. We can reduce the impact of such rays by reducing the weights or calculating the corresponding points using features with higher uniqueness instead of color. The figure on the right shows the overlaid rendered and observed images at each optimization iteration. The overlapped object area between the rendered and observed images is small for iter=0. Optimization using only photometric error gives little progress since it cannot provide a reasonable gradient. Rough alignment using the reprojection error by the 100th iteration enables the use of photometric error with an effective gradient, which provides highly accurate localization.

6 Conclusions

This study proposed NeDDF to represent reciprocally constrained distance and density fields. We extended the distance field to a formulation that can adapt to any density field. We also derived the conversion formula between distance and density using the distance and its gradient, enabling learning the these fields while constraining each other. We also alleviate the problem of discontinuity points by introducing the auxiliary gradient. The visualization experiments demonstrated that NeDDF could acquire the properties of both the conventional density field and the distance field. The quantitative evaluation showed that NeDDF could provide competitive quality of novel view synthesis, more stable meshes, and a more comprehensive range of following camera poses than NeRF.

One limitation of our method is the lack of information about the distance field inside the objects. Since NeDDF is based on UDF, it cannot provide helpful gradient directions in the interior object region. NeDDF also has the same limitations as the original NeRF [14], such as time-consuming optimization and rendering. However, since NeDDF provides a differentiable density field and retains the same formulation as NeRF, many of the latest advances in improving NeRF, such as speedup and stabilization, may apply to NeDDF. In addition, we calculated the pseudo correspondence points from the colors in the localization as a simple example of obtaining the reprojection error. We believe that using information with higher uniqueness, such as semantic segmentation, will increase the usefulness of the reprojection error.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP19H00806, JP21KK0070 and JP22H01580. We would like to thank Mr. Towaki Takikawa, Mr. Naoya Chiba, and Mr. Ryota Suzuki for their helpful discussions.

References

- [1] Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2565–2574 (2020)

- [2] Chen, H.H., Huang, T.S.: A survey of construction and manipulation of octrees. *Computer Vision, Graphics, and Image Processing* **43**(3), 409–431 (1988)
- [3] Chibane, J., Mir, A., Pons-Moll, G.: Neural unsigned distance fields for implicit function learning. In: *Advances in Neural Information Processing Systems (NeurIPS)*. pp. 21638–21652 (2020)
- [4] Diederik P. Kingma, J.B.: Adam: A method for stochastic optimization. In: *International Conference on Learning Representations* (2015)
- [5] Ge, L., Liang, H., Yuan, J., Thalmann, D.: 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 1991–2000 (2017)
- [6] Groueix, T., Fisher, M., Kim, V.G., Russell, B., Aubry, M.: AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 216–224 (2018)
- [7] Laine, S., Karras, T.: Efficient sparse voxel octrees. *IEEE Transactions on Visualization and Computer Graphics* **17**(8), 1048–1059 (2010)
- [8] Lee, J.H., Ha, H., Dong, Y., Tong, X., Kim, M.H.: Texturefusion: High-quality texture acquisition for real-time rgbd scanning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 1272–1280 (2020)
- [9] Li, Y., Bozic, A., Zhang, T., Ji, Y., Harada, T., Nießner, M.: Learning to optimize non-rigid tracking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4910–4918 (2020)
- [10] Li, Y., Zhang, T., Nakamura, Y., Harada, T.: Splitfusion: Simultaneous tracking and mapping for non-rigid scenes. In: *IROS*. pp. 5128–5134 (2020)
- [11] Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 6498–6508 (2021)
- [12] Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. pp. 5741–5751 (2021)
- [13] Liu, X., Di, X.: Tanhexp: A smooth activation function with high convergence speed for lightweight neural networks. *arXiv preprint arXiv:2003.09855* (2020)
- [14] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 405–421 (2020)
- [15] Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 343–352 (2015)
- [16] Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: *10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. pp. 127–136 (2011)
- [17] Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)* **32**(6), 1–11 (2013)
- [18] Oechsle, M., Peng, S., Geiger, A.: Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. pp. 5589–5599 (2021)
- [19] Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 165–174 (2019)
- [20] Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* pp. 5865–5874 (2021)
- [21] Park, K., Sinha, U., Hedman, P., Barron, J.T., Bouaziz, S., Goldman, D.B., Martin-Brualla, R., Seitz, S.M.: Hyper-nerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (ToG)* **40**(6), 1–12 (2021)

- [22] Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10318–10327 (2021)
- [23] Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 12959–12970 (2021)
- [24] Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
- [25] Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: NeRF—: Neural radiance fields without known camera parameters. arXiv preprint arXiv: 2102.07064 (2021)
- [26] Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., Tombari, F., Tompkin, J., Sitzmann, V., Sridhar, S.: Neural fields in visual computing and beyond. arXiv preprint arXiv: 2111.11426 (2021)
- [27] Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 206–215 (2018)
- [28] Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
- [29] Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 33 (2020)
- [30] Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: iNeRF: Inverting neural radiance fields for pose estimation. In: Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS). pp. 1323–1330 (2021)

Appendix

7 Architecture details

Fig. 12 shows the details of our architecture. The network uses multilayer perceptrons (MLPs) with a width of 256, the same as in NeRF [14] and NeuS [24]. The network calculates the density using the first derivative value for the output distance D . This calculation requires careful setup of the Positional Encoding (PE) and activation functions. Using an objective function for the density field requires gradients up to the second derivative for activation functions. The architecture uses tanhExp [13] as the activation function whose second derivative is continuous. In the conventional method [14], the PE up to L dimensions utilizes values such as the following equation:

$$\gamma(\mathbf{p}) = [\sin(\mathbf{p}), \cos(\mathbf{p}), \dots, \sin(2^{L-1}\mathbf{p}), \cos(2^{L-1}\mathbf{p})]^T. \quad (19)$$

It is the concatenation of the sin and cos of each dimension of position \mathbf{p} scaled by powers of 2 from 1 to $2L - 1$. This PE amplifies by the frequency in the hierarchy of first derivatives, which emphasizes the high-frequency elements in the density field. In other words, the maximum and minimum frequency components have a scale difference of 2^{L-1} in their influence on the density derivative, thus making the learning process unstable. Therefore, our architecture damps the high-frequency element so that the scale in the single-differentiation hierarchy is the same as the original PE, as shown in the following equation:

$$\gamma'(\mathbf{p}) = \left[\sin(\mathbf{p}), \cos(\mathbf{p}), \dots, \frac{1}{2^{L-1}}\sin(2^{L-1}\mathbf{p}), \frac{1}{2^{L-1}}\cos(2^{L-1}\mathbf{p}) \right]^T. \quad (20)$$

However, since the PE neglects the high-frequency component in the non-derivative hierarchy, we need to add an intermediate input of the conventional $\gamma(\mathbf{p})$ in the layers after the distance output for learning detailed color fields. Note that the performance of restoring the high-frequency component of the color field is worse than that of NeRF for the same network size.

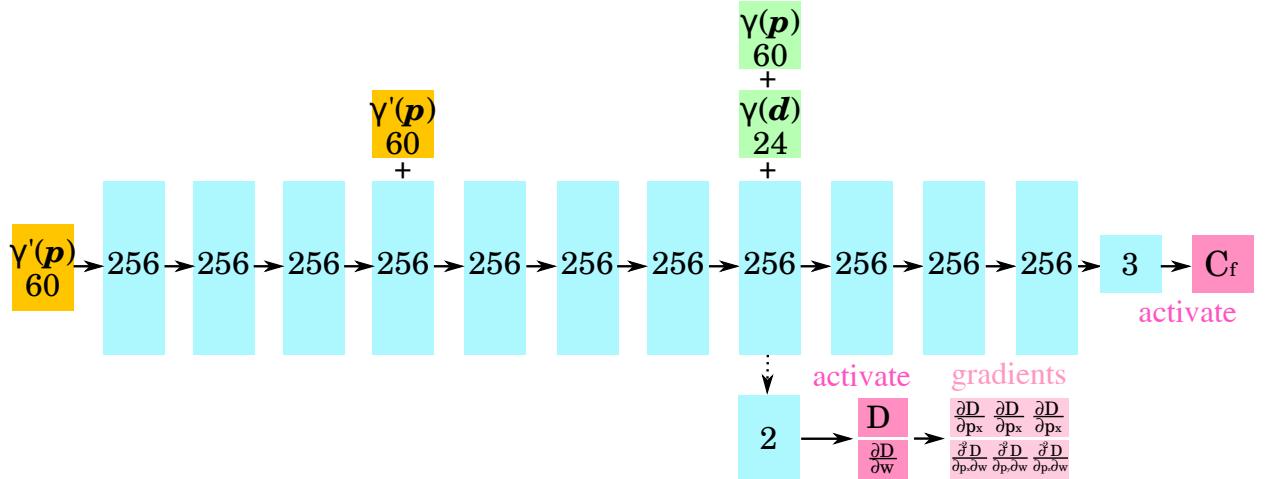


Figure 12: Details of network architecture in NeDDF

8 Deriving the conversion from distance to density

This section describes the derivation details of Equation 10 in the main paper. For the distance field around position $\mathbf{p} \in \mathbb{R}^3$, we consider $D(\mathbf{r}(t))$, $\mathbf{r}(t) = \mathbf{p} + tv$, which is sliced in the gradient direction \mathbf{v} . Calculating the derivative of the distance field in the direction of the gradient, $\frac{\partial D}{\partial t}$, we can derive an expression for σ as follows:

$$\frac{\partial D(\mathbf{r}(t))}{\partial t}|_{t=0} \quad (21)$$

$$= \lim_{\Delta t \rightarrow 0} \frac{d(\mathbf{r}(\Delta t), \mathbf{v}) - d(\mathbf{r}(0), \mathbf{v})}{\Delta t}. \quad (22)$$

The first term of Equation 22 can be deformed as follows:

$$d(\mathbf{r}(\Delta t), \mathbf{v}) \quad (23)$$

$$= \int_{t_n}^{t_f} \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}(s + \Delta t))ds\right) \sigma(\mathbf{r}(t + \Delta t)) dt \quad (24)$$

$$= \int_{t_n}^{t_f} \exp\left(-\int_{t_n + \Delta t}^{t + \Delta t} \sigma(\mathbf{r}(s))ds\right) \sigma(\mathbf{r}(t + \Delta t)) dt. \quad (25)$$

We set $S(t_n, t) := \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}(s))ds\right) \sigma(\mathbf{r}(t))dt$. When t_f takes a sufficiently large value until all of the light gets reflected, the Equations 26, 27, and 28 are valid.

$$T(t_f) = 0 \quad (26)$$

$$S(t_n, t_f) = 0 \quad (27)$$

$$\int_{t_n}^{t_f} S(t_n, t_f) = 1 \quad (28)$$

With Equations 26, 27, and 28, $d(\mathbf{r}(\Delta t), \mathbf{v})$ can be deformed as follows:

$$d(\mathbf{r}(\Delta t), \mathbf{v}) \quad (29)$$

$$= \int_{t_n}^{t_f} S(t_n + \Delta t, t + \Delta t) dt \quad (30)$$

$$= \int_{t_n}^{t_f} S(t_n + \Delta t, t + \Delta t)(t + \Delta t) dt - \int_{t_n}^{t_f} S(t_n + \Delta t, t + \Delta t) \Delta t dt \quad (31)$$

$$= \int_{t_n + \Delta t}^{t_f + \Delta t} S(t_n + \Delta t, t) dt - \Delta t \int_{t_n}^{t_f} S(t_n + \Delta t, t + \Delta t) dt \quad (32)$$

$$= \int_{t_n}^{t_f} S(t_n + \Delta t, t) dt + \int_{t_f}^{t_n + \Delta t} S(t_n + \Delta t, t) dt \\ - \int_{t_n}^{t_n + \Delta t} S(t_n + \Delta t, t) dt - \Delta t. \quad (33)$$

We calculate the first term of Equation 33 as follows:

$$\int_{t_n}^{t_f} S(t_n + \Delta t, t) dt \quad (34)$$

$$= \int_{t_n}^{t_f} \exp\left(-\int_{t_n + \Delta t}^t \sigma(\mathbf{r}(s))ds\right) \sigma(\mathbf{r}(t)) dt \quad (35)$$

$$= \int_{t_n}^{t_f} T(t) T(t_n + \Delta t) \sigma(t_n + \mathbf{r}(t)) dt \quad (36)$$

$$= T(t_n + \Delta t) d(\mathbf{r}(0), \mathbf{v}). \quad (37)$$

Using Equation 27, the second term of Equation 33 is equal to 0. The third term of Equation 33 is calculated as follows:

$$\int_{t_n}^{t_n + \Delta t} S(t_n + \Delta t, t) dt \quad (38)$$

$$= T(t_n + \Delta t) \int_{t_n}^{t_n + \Delta t} T(t) \sigma(\mathbf{r}(t)) dt. \quad (39)$$

Same as in the main paper, we assume that t_n is small enough to be valid at $T(t_n) = 1$. Equation 39 converges to $\Delta t \sigma(\mathbf{r}(t_n)) t_n$ as $\Delta t \rightarrow 0$. Therefore, the Equation 22 can be deformed as follows:

$$\frac{\partial D(\mathbf{r}(t))}{\partial t} \Big|_{t=0} \quad (40)$$

$$= \lim_{\Delta t \rightarrow 0} \frac{d(\mathbf{r}(\Delta t), \mathbf{v}) - d(\mathbf{r}(0), \mathbf{v})}{\Delta t} \quad (41)$$

$$= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} [T(t_n + \Delta t)d(\mathbf{r}(0), \mathbf{v}) - \Delta t \sigma(\mathbf{r}(t))t_n - \Delta t - d(\mathbf{r}(0), \mathbf{v})] \quad (42)$$

$$= \lim_{\Delta t \rightarrow 0} \left[\frac{T(t_n + \Delta t) - 1}{\Delta t} d(\mathbf{r}(0), \mathbf{v}) - \sigma(\mathbf{r}(t_n))t_n - 1 \right] \quad (43)$$

$$= \lim_{\Delta t \rightarrow 0} \left[\frac{\exp(\Delta t \sigma(\mathbf{r}(t_n))) - \exp(0)}{\Delta t} d(\mathbf{r}(0), \mathbf{v}) - \sigma(\mathbf{r}(t_n))t_n - 1 \right] \quad (44)$$

$$= \sigma(\mathbf{r}(t_n))d(\mathbf{r}(0), \mathbf{v}) - \sigma(\mathbf{r}(t_n))t_n - 1 \quad (45)$$

$$= -1 + (D(\mathbf{r}(0)) - t_n) \sigma(\mathbf{r}(t_n)). \quad (46)$$

9 Parameter selection for the shape of the auxiliary gradient

Equation 15 in the main paper is a penalty term that constrains the shape of the auxiliary gradient by the hyperparameter α . Fig. 13 shows that the auxiliary gradient becomes active in a narrower range than the distance field when $\alpha \leq 1$, and the shape becomes more concentrated near the cusps as α is larger. This constraint leads to a unique set of auxiliary gradients.

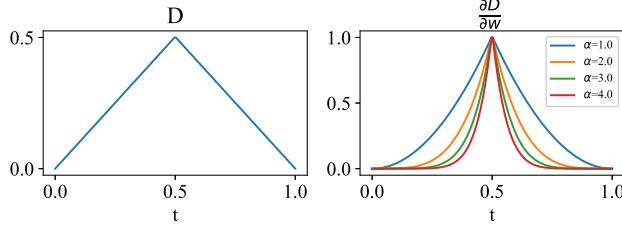


Figure 13: Shape of auxiliary gradient for each α

Fig. 14 is a colorized visualization of the distance field, density field, and auxiliary gradient in the 2D slice. We can see that the auxiliary gradient becomes strongly activated near the cusp of the distance field, where the distances from several objects are similar. Fig. 15 shows the difference in rendering results with and without auxiliary gradients. Without auxiliary gradients, incorrect volume densities occur in the empty region.

10 Evaluation of reconstruction performance in smoke scenes

Most of the previous methods do not provide benchmarks for scenes with subjects containing smoke. We produce a synthetic dataset for a smoke-subject scene where the density varies over a wide area, and we qualitatively evaluate the proposed method's performance. As with the nerf synthetic dataset [14], the dataset consists of 100 viewpoints each in a hemispherical plane for train/valid data and 200 viewpoints in orbit for test data. For each shot, we record RGB and Transmittance information at a resolution of 800×800 .

Fig. 16 shows the rendered image from the test viewpoint. Our method achieves high-quality Novel View Synthesis even in smoke-like scenes. Fig. 17 also shows the visualization results of the slices for the distance field, density field, and auxiliary gradient. Since the distance field shows that the distance increases again inside the object, the field's estimation also holds true inside the object. In addition, the minima of the distance field are larger than those in Fig. 14, indicating that the assumption of expressing low density with large minima of the distance field works reasonably. The density field actually expresses a translucent state rather than a bipolar one.

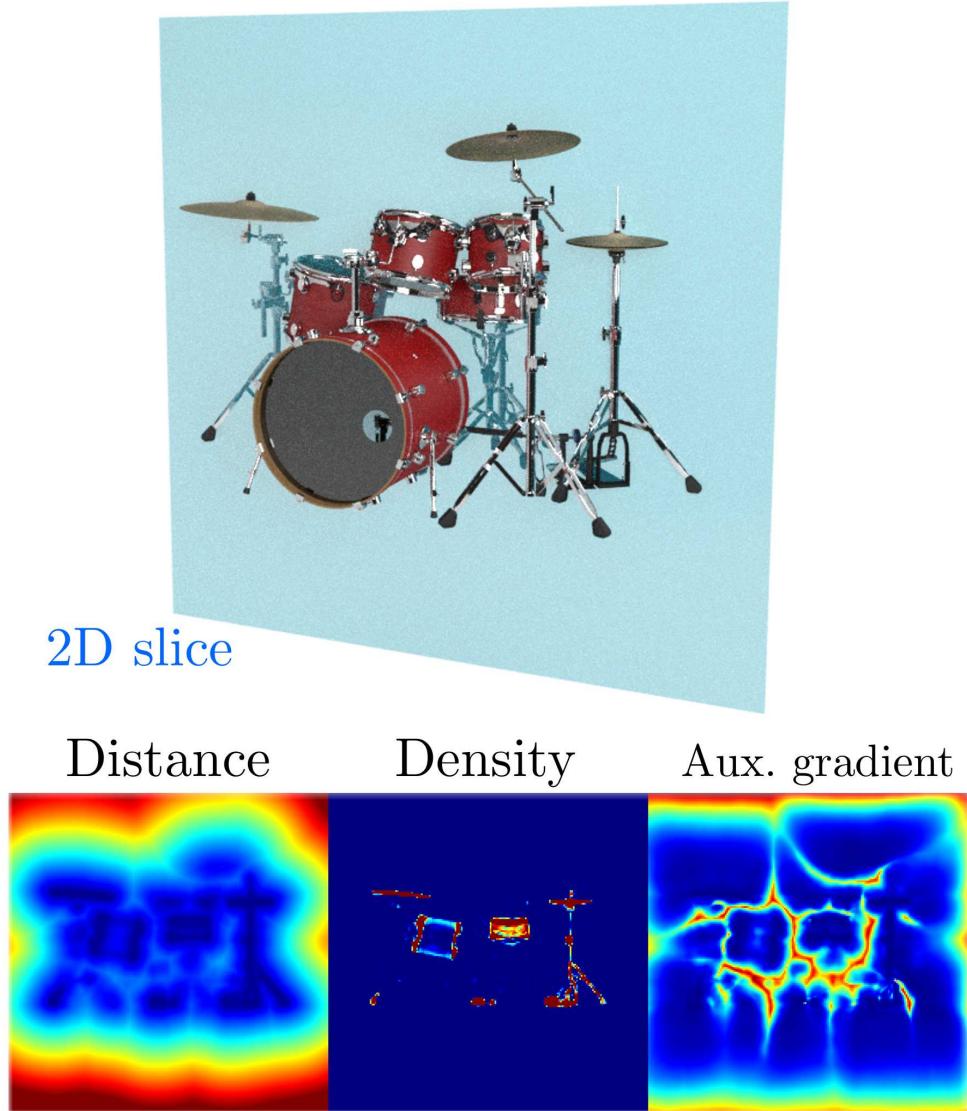


Figure 14: Visualization of the 2D slice for distance, density, and auxiliary gradient.

11 Evaluation of localization performance in other scenes

For other scenes in the NeRF synthetic dataset [14], we verify the camera localization performance in the same way as in experiment (b) of main paper. Fig. 18 plots the number of camera postures for which the position and angular errors are lower than the threshold values for each scene. In all cases, the use of the reprojection error improves performance more than the use of the photometric error alone, as in iNeRF[30]. Even in cases such as Drums and Ficus scenes, where the optimization result with only the reprojection error is worse than the initial value, we can see an improvement of performance due to the increase in the common area of the field of view. On the other hand, in scenes where the uniqueness of color information is not sufficient, such as the Materials scene, reprojection error may degrade performance due to mismatches between corresponding points. In the case where many local solutions for photometric errors exist, such as Mic and Ship scenes, the use of reprojection error does not avoid the local solutions and does not improve the performance. We believe that we can improve such scenes by propagating unique features other than color to the empty regions in the same way as color fields.

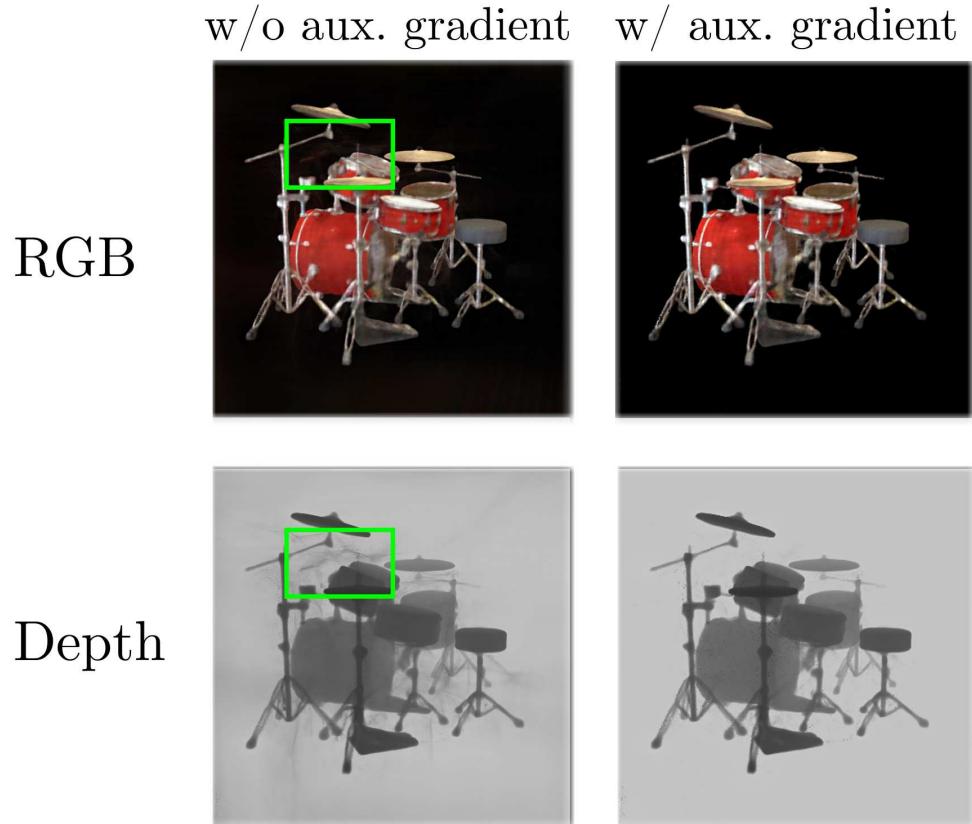


Figure 15: The difference in rendering results with and without auxiliary gradients.

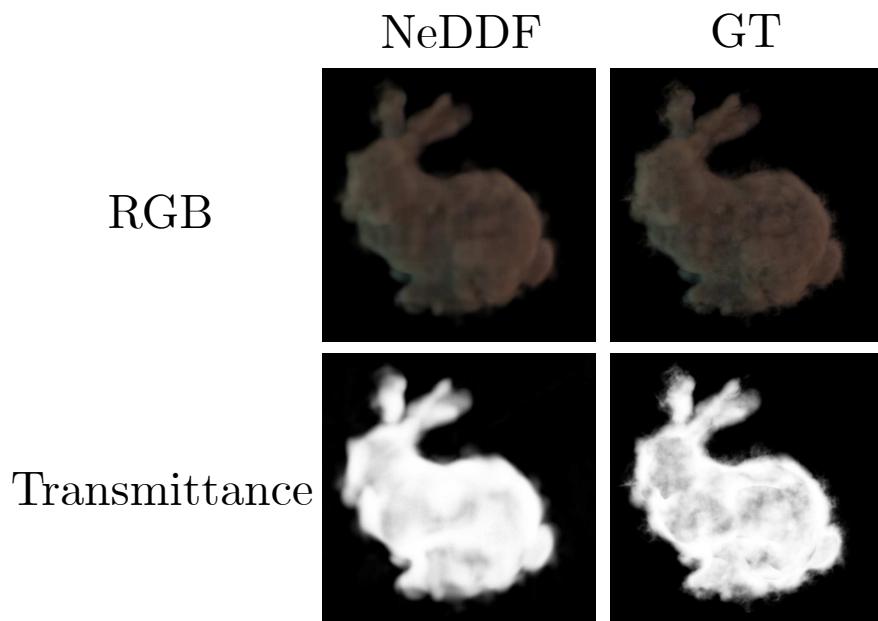


Figure 16: Rendering results in smoke scene

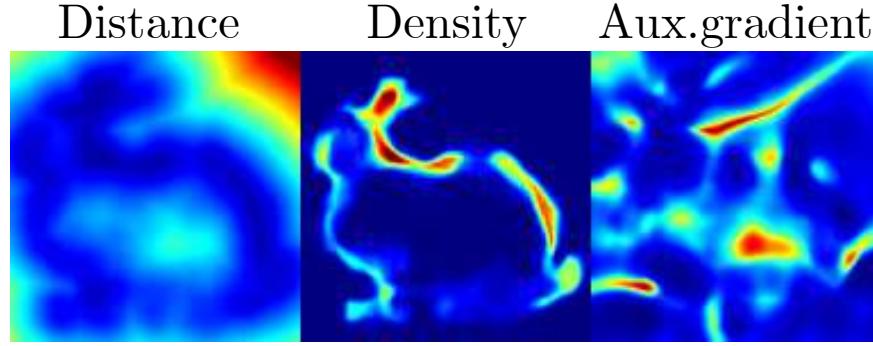


Figure 17: Visualization of the 2D slice for distance, density, and auxiliary gradient in smoke scene.

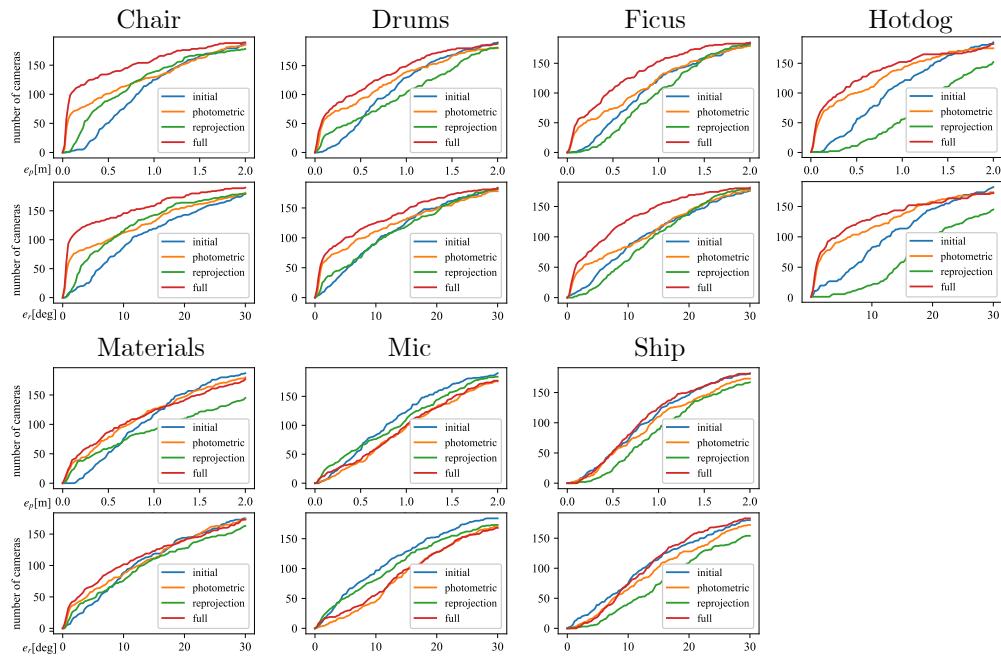


Figure 18: Quantitative evaluation of camera poses estimation accuracy in other scenes. The horizontal axis represents the position and angle error, and the vertical axis represents the number of cameras recovered under the errors.