

# Learning Scene-Level Signed Directional Distance Function for Aerial Autonomy

Zhirui Dai

Hojoon Shin

Yulun Tian

Ki Myung Brian Lee

Nikolay Atanasov

**Abstract**—Dense differentiable environment representations are critical for navigation and exploration by aerial robots. In this work, we explore a novel implicit scene representation, the Signed Directional Distance Function (SDDF), to enhance geometry modeling and differentiable trajectory optimization. Unlike signed distance function (SDF) and similar to neural radiance fields (NeRF), SDDF has a position and viewing direction as input. Like SDF and unlike NeRF, SDDF directly provides distance to the observed surface along the viewing direction, allowing efficient view synthesis without iterative ray marching. To learn and predict scene-level SDDF efficiently, we develop a differentiable hybrid representation that combines explicit ellipsoid priors and implicit neural residuals. This approach allows the model to effectively handle large distance discontinuities around obstacle boundaries while preserving the ability for dense high-fidelity prediction. We show that SDDF is competitive with the state-of-the-art neural implicit scene models in terms of reconstruction accuracy and rendering efficiency, while allowing differentiable view prediction for robot trajectory optimization.

## I. INTRODUCTION

Aerial robots are increasingly deployed in a priori unknown and unstructured environments. Successful operations require efficient representation and prediction of environment geometry from sensor observations to support collision checking for safe navigation, occlusion prediction for autonomous exploration, or grasp pose generation for aerial manipulation. While conventional explicit representations such as meshes, point clouds, and voxels are well established, they are not continuous and does not support differentiation, a key requirement for navigation and trajectory optimization. Recent work has focused on implicit and differentiable scene representations that use neural fields to model occupancy [1], signed distance function (SDF) [2], and radiance field [3]. Although these implicit methods offer superior fidelity, they require multiple network forward passes, complicated calculations per pixel/ray, and high memory usage, posing significant challenges for deployment onboard resource-constrained aerial robots.

To overcome these limitations, we propose and investigate a novel implicit representation, called the *Signed Directional Distance Function* (SDDF). The SDDF takes query position and viewing direction as input and directly output the distance to the observed surface along the viewing direction. By learning a geometry representation in the space of positions and directions, SDDF support arbitrary view synthesis and efficient occlusion queries in a feedforward manner, eliminating the need for iterative sphere tracing required for SDF. Further, SDDF can be trained purely from range observations such as

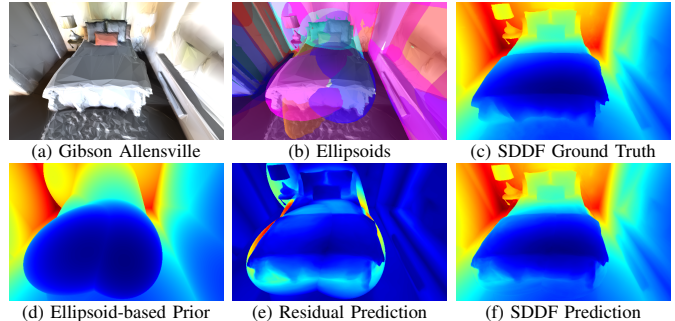


Fig. 1: (a), (c): Scene-level signed directional distance function (SDDF). (a), (b), (d): Our method uses ellipsoids as an initial coarse approximation of the shapes of objects in the environment. (e), (f): The ellipsoid prior is refined by a latent feature network and a shared decoder to predict the surface reconstruction residual.

depth images and LiDAR scans, unlike NeRF [3] and Gaussian Splatting (GS) [4] that require photometric supervision.

However, learning SDDF is challenging due to the incorporation of direction in the input space and discontinuities caused by occlusions. For these reasons, previous methods that study similar formulations [5, 6] are limited to single object shape modeling. In contrast, we develop a *scene-level* SDDF representation that is better suited for aerial autonomy. To this end, we design a hybrid explicit-implicit model that combines an ellipsoid-based prior and an implicit neural residual network to approximate the SDDF in a differentiable way (Fig. 1). Our experiments show that our method is competitive with state-of-the-art SDF, GS, and NeRF in reconstruction performance while supporting efficient differentiable view optimization.

## II. LEARNING SCENE-LEVEL SDDF

To enable efficient view optimization for UAVs equipped with range sensors (LiDARs or depth cameras), we need to learn an environment model that is capable of efficient and differentiable synthesis of arbitrary distance views. Let the occupied space in the environment be represented by a set  $\mathcal{O} \subset \mathbb{R}^3$ . Consider a set of measurements  $\{\mathbf{T}_t, \mathcal{Z}_t\}_{t=1}^T$ , where  $\mathbf{T}_t \in SE(3)$  is the sensor pose at time  $t$  and  $\mathcal{Z}_t = \{\mathbf{v}_i, r_{t,i}\}_{i=1}^N$  are the  $N$  viewing directions  $\mathbf{v}_i \in \mathbb{S}^2$  and corresponding range measurements  $r_{t,i} \in \mathbb{R}_{>0}$ . Our objective is to learn a representation of the occupied space  $\mathcal{O}$  in the form of a signed directional distance function.

**Definition 1.** The *signed directional distance function* (SDDF) of a set  $\mathcal{O} \subset \mathbb{R}^n$  is a function  $f : \mathbb{R}^n \times \mathbb{S}^{n-1} \rightarrow \mathbb{R} \cup \{\pm\infty\}$  that measures the signed distance from a point  $\mathbf{p} \in \mathbb{R}^n$  to the set boundary  $\partial\mathcal{O}$  along a direction  $\mathbf{v} \in \mathbb{S}^{n-1}$ , defined as:

$$f(\mathbf{p}, \mathbf{v}; \mathcal{O}) := \begin{cases} \min\{d > 0 \mid \mathbf{p} + d\mathbf{v} \in \partial\mathcal{O}\}, & \mathbf{p} \notin \mathcal{O}, \\ \max\{d \leq 0 \mid \mathbf{p} + d\mathbf{v} \in \partial\mathcal{O}\}, & \mathbf{p} \in \mathcal{O}. \end{cases} \quad (1)$$

The authors are with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093 USA. Email: {zhidai, hoshin, yut034, kmlblee, natanasov}@ucsd.edu.

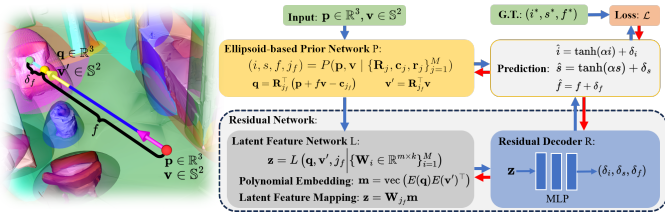


Fig. 2: Method overview. Blue arrows show the data flow in the forward pass, and red arrows represent the backward pass.

Intuitively, the SDDF can be understood as a directional formulation of the SDF. It is well known [7, 8] that SDFs satisfy an Eikonal equation  $\|\nabla_{\mathbf{p}} f_{\text{SDF}}(\mathbf{p}; \mathcal{O})\|_2 = 1$ , which is useful for regularizing or designing the structure of models for estimating SDF. Similarly, SDDF satisfies a directional Eikonal equation  $\mathbf{v}^\top \nabla_{\mathbf{p}} f(\mathbf{p}, \mathbf{v}; \mathcal{O}) = -1$ , a property that we will encode in the design of our network architecture.

As shown in Fig. 2, our model combines both explicit and implicit representations. First, an explicit ellipsoid-based Prior network  $P(\mathbf{p}, \mathbf{v})$  is introduced to predict a coarse SDDF prior  $f(\mathbf{p}, \mathbf{v})$ . Then, a residual network consisting of a Latent feature network  $L$  and a Residual decoder  $R$  are used to predict an SDDF correction  $\delta_f(\mathbf{p}, \mathbf{v})$ , so that the combination accurately models the true SDDF  $f^*(\mathbf{p}, \mathbf{v})$  as  $\hat{f}(\mathbf{p}, \mathbf{v}) = f(\mathbf{p}, \mathbf{v}) + \delta_f(\mathbf{p}, \mathbf{v})$ .

### A. Ellipsoid-based Prior Network

To take advantage of an explicit representation for occlusion modeling, we design an ellipsoid-based prior network  $P$ , which uses a set of ellipsoids to approximate the structure of the environment based on range measurements and leaves the task of learning fine details to the residual network  $R$ .

First, for simplicity, consider a single ellipsoid given by  $\mathcal{E} = \{\mathbf{y} \in \mathbb{R}^3 \mid (\mathbf{y} - \mathbf{c})^\top \mathbf{R} \mathbf{Q}_0^{-2} \mathbf{R}^\top (\mathbf{y} - \mathbf{c}) \leq 1\}$ , where  $\mathbf{c} \in \mathbb{R}^3$  and  $\mathbf{R} \in SO(3)$  are the position and orientation,  $\mathbf{Q}_0 = \text{diag}(\mathbf{r})$ , and  $\mathbf{r} \in \mathbb{R}_+^3$  are the radii of the ellipsoid. Then, the SDDF prior of a single ellipsoid  $\mathcal{E}$  is defined as

$$f(\mathbf{p}, \mathbf{v}; \mathcal{E}) = \begin{cases} -\frac{\det \mathbf{Q}_0 \sqrt{\beta} + \mathbf{p}'^\top \mathbf{Q}_1^2 \mathbf{v}'}{\mathbf{v}'^\top \mathbf{Q}_1^2 \mathbf{v}'}, & v(\mathbf{p}, \mathbf{v}) \geq 0, \\ \infty, & v(\mathbf{p}, \mathbf{v}) < 0, \end{cases} \quad (2)$$

where  $\beta = \max(i(\mathbf{p}, \mathbf{v}), 0) + \epsilon$ ,  $\epsilon > 0$  is a small value introduced for the numerical stability.  $i(\mathbf{p}, \mathbf{v}) = \mathbf{v}'^\top \mathbf{Q}_1^2 \mathbf{v}' - \mathbf{w}'^\top \mathbf{Q}_0^2 \mathbf{w}'$  is the intersection indicator that  $i(\mathbf{p}, \mathbf{v}) \geq 0$  when the ray intersects the ellipsoid.  $\mathbf{Q}_1 = \det(\mathbf{Q}_0) \mathbf{Q}_0^{-1}$ ,  $\mathbf{p}' = \mathbf{R}^\top (\mathbf{p} - \mathbf{c})$ ,  $\mathbf{v}' = \mathbf{R}^\top \mathbf{v}$ , and  $\mathbf{w}' = \mathbf{p}' \times \mathbf{v}'$ . The validity indicator function is defined as:

$$v(\mathbf{p}, \mathbf{v}) = -\frac{\det \mathbf{Q}_0 \sqrt{\beta} + \mathbf{p}'^\top \mathbf{Q}_1^2 \mathbf{v}'}{\mathbf{v}'^\top \mathbf{Q}_1^2 \mathbf{v}'} s(\mathbf{p}, \mathbf{v}), \quad (3)$$

where  $s(\mathbf{p}, \mathbf{v}) = \mathbf{p}'^\top \mathbf{Q}_1^2 \mathbf{p}' - \det \mathbf{Q}_0^2$  is the sign indicator. When  $\mathbf{p}$  is outside of the ellipsoid,  $s(\mathbf{p}, \mathbf{v}) > 0$ . Fig. 3 shows the ellipsoid SDDF prior  $f(\mathbf{p}, \mathbf{v}; \mathcal{E})$  for a 2D example.

To model scenes with multiple objects at different locations, we consider a set of  $M$  ellipsoids  $\mathcal{E}_j$  for  $1 \leq j \leq M$ . The SDDF of a union of ellipsoids is the minimum of the individual

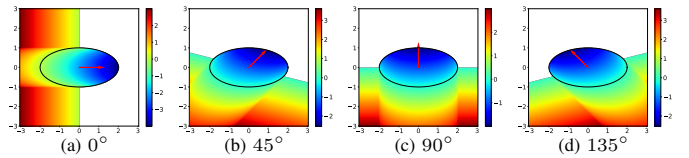


Fig. 3: 2D visualization of the single ellipsoid SDDF  $f(\mathbf{p}, \mathbf{v}; \mathcal{E})$  in (2) for fixed  $\mathbf{v}$  and varying  $\mathbf{p}$ .

ellipsoid SDDFs but with the intersected ellipsoids prioritized:

$$f(\mathbf{p}, \mathbf{v}; \cup_j \mathcal{E}_j) = \begin{cases} \min_{j: i_j(\mathbf{p}, \mathbf{v}) \geq 0} f_j(\mathbf{p}, \mathbf{v}), & \exists i_j(\mathbf{p}, \mathbf{v}) \geq 0, \\ \min_j f_j(\mathbf{p}, \mathbf{v}), & \text{otherwise.} \end{cases} \quad (4)$$

Thus, given  $M$  ellipsoids with pose and radii  $\mathbf{R}_j, \mathbf{c}_j, \mathbf{r}_j$  for  $1 \leq j \leq M$ , we define our overall ellipsoid-based prior as:

$$(i, s, f, j_f) = P(\mathbf{p}, \mathbf{v} \mid \{\mathbf{R}_j, \mathbf{c}_j, \mathbf{r}_j\}_{j=1}^M), \quad (5)$$

where  $i = \max_j i_j(\mathbf{p}, \mathbf{v})$ ,  $s = \min_j s_j(\mathbf{p}, \mathbf{v})$ , and  $j_f$  is the index of the ellipsoid selected by (4) to be used for subsequent residual calculation. By construction, the ellipsoid-based prior satisfies the Eikonal equation.

### B. Residual Network

The network  $P$  provides a coarse geometric prior but does not yield accurate predictions. We design a residual network to predict a correction term  $\delta_f(\mathbf{p}, \mathbf{v})$  so that the SDDF prediction of our combined prior and residual,  $\hat{f}(\mathbf{p}, \mathbf{v}) = f(\mathbf{p}, \mathbf{v}) + \delta_f(\mathbf{p}, \mathbf{v})$ , is accurate.

First, given the intersected ellipsoid  $\mathcal{E}_{j_f}$  selected by (5), we obtain the intersection point  $\mathbf{q}$  in the ellipsoid frame:  $\mathbf{q} = \mathbf{R}_{j_f}^\top (\mathbf{p} + \mathbf{f}\mathbf{v} - \mathbf{c}_{j_f}) = \mathbf{p}' + \mathbf{f}\mathbf{v}'$ . Then, we train a latent feature network  $\mathbf{z} = L(\mathbf{q}, \mathbf{v}', j_f)$  with the intersection point  $\mathbf{q}$ , local viewing direction  $\mathbf{v}'$ , and ellipsoid index  $j_f$  as input and a latent feature  $\mathbf{z}$  as output:

$$\mathbf{z} = L(\mathbf{q}, \mathbf{v}', j_f \mid \{\mathbf{W}_i\}_{i=1}^M) = \mathbf{W}_{j_f} \mathbf{m} \in \mathbb{R}^m, \quad (6)$$

$$\mathbf{m} = \text{vec}(E(\mathbf{q})E(\mathbf{v}')^\top) \in \mathbb{R}^{100}, \quad (7)$$

$$E(\mathbf{p}) = [p_x^2, p_x p_y, p_x p_z, p_y^2, p_y p_z, p_z^2, p_x, p_y, p_z, 1]^\top, \quad (8)$$

where  $E: \mathbb{R}^3 \rightarrow \mathbb{R}^{10}$  is a degree-2 monomial embedding,  $\text{vec}(\cdot)$  concatenates the columns of the input matrix,  $\mathbf{m}$  is a vector of degree-2 monomials, and  $\mathbf{W}_i \in \mathbb{R}^{m \times 100}$ .

The latent feature vector  $\mathbf{z} \in \mathbb{R}^m$  is then decoded by the residual decoder, which is a multi-layer perceptron  $R: \mathbb{R}^m \rightarrow \mathbb{R}^3$ , into three residual predictions  $(\delta_i, \delta_s, \delta_f)$ . Then, the final predictions of the SDDF value is:

$$\hat{f}(\mathbf{p}, \mathbf{v}) = f(\mathbf{p}, \mathbf{v}) + \delta_f(\mathbf{p}, \mathbf{v}), \quad (9)$$

where  $\alpha > 0$  is a hyperparameter. In addition, we also get the final prediction of the intersection indicator  $\hat{i}(\mathbf{p}, \mathbf{v}) = \tanh(\alpha i(\mathbf{p}, \mathbf{v})) + \delta_i(\mathbf{p}, \mathbf{v})$  and the sign indicator  $\hat{s}(\mathbf{p}, \mathbf{v}) = \tanh(\alpha s(\mathbf{p}, \mathbf{v})) + \delta_s(\mathbf{p}, \mathbf{v})$ .

And nicely, the joint prior-residual SDDF prediction  $\hat{f}$  in (9) still satisfies the SDDF directional Eikonal equation by construction, which is discussed in details in our extended paper [9]. Because our SDDF model satisfies the Eikonal equation by construction, we do not need an extra loss term to regularize the network and can use fewer parameters in the model, making it more efficient to train.

### III. APPLICATION TO VIEWPOINT OPTIMIZATION

Our SDDF model is differentiable and, hence, enables continuous viewpoint optimization, which can be used, for example, to explore an unknown environment. For simplicity, we first consider determining the next-best view and then scale up to optimization of a trajectory of several views.

Our SDDF model can predict a point cloud measurement from any desired sensor pose  $(\mathbf{p}_t, \mathbf{R}_t)$  as:

$$\mathcal{P}_t = \{\mathbf{p}_t + \hat{f}_i \mathbf{R}_t \mathbf{v}_i\}_{i=1}^N, \quad (10)$$

where  $\{\mathbf{v}_i\}_{i=1}^N$  are the ray directions in the sensor frame and  $\hat{f}_i$  are the SDDF predictions for each ray. The utility of a point-cloud measurement for the purpose of exploration or environment coverage can be evaluated in terms of the visible region volume. We use the following loss to measure the (negative) size of the visible volume:

$$\mathcal{L}_v(\{\hat{f}_i\}_{i=1}^N) = -\frac{1}{2N} \sum_{i=1}^N (\max\{\hat{f}_i, 0\})^2. \quad (11)$$

Often, it is also desirable to design consecutive views  $\mathcal{P}_t$  and  $\mathcal{P}_{t+1}$  to have small overlap in order to observe a larger overall area. We encode this using the following overlap loss:

$$\mathcal{L}_o(\mathcal{P}_t, \mathcal{P}_{t+1}) = -\frac{\sum_{\mathbf{p} \in \mathcal{P}_t, \mathbf{q} \in \mathcal{P}_{t+1}} \min\{\|\mathbf{p} - \mathbf{q}\|_2, d_{\max}\}}{|\mathcal{P}_t| |\mathcal{P}_{t+1}|}, \quad (12)$$

where  $d_{\max} > 0$  is a distance threshold of no overlap.

Additionally, we must also ensure that the sensor is not in collision with any obstacles. To do so, we introduce a set of risk detection rays, which are uniformly sampled from the sphere that contains the robot, and obtain their SDDF predictions  $\{\hat{f}_i^r\}_{i=1}^M$ . The risk loss is defined as:

$$\mathcal{L}_r(\{\hat{f}_i^r\}_{i=1}^M) = \frac{1}{M} \sum_{i=1}^M \max\{d_{\text{safe}} - \hat{f}_i^r, 0\}, \quad (13)$$

where  $d_{\text{safe}} > 0$  is a safe distance threshold, chosen such that  $\hat{f}_i^r < d_{\text{safe}}$  implies a potential collision.

We optimize the camera pose  $(\mathbf{p}_{t+1}, \mathbf{R}_{t+1})$  at time  $t + 1$  by minimizing the weighted sum  $\mathcal{L} = w_o \mathcal{L}_o + w_v \mathcal{L}_v + w_r \mathcal{L}_r$ , where  $w_o$ ,  $w_v$ , and  $w_r$  are weights.

There are many ways to scale this method up to viewpoint trajectory optimization. It is inefficient to optimize every pose on the trajectory because two views that are close to each other are very likely to overlap significantly, and the robot kinematic constraints may not allow much adjustment for the viewpoint. Therefore, we suggest optimizing certain waypoints on the trajectory. We incrementally optimize  $n$  poses  $\{\mathbf{p}_i, \mathbf{R}_i\}_{i=0}^n$  generated by an off-the-shelf planning algorithm, such as RRT\* [10]. Specifically, we refine each pose  $\{\mathbf{p}_i, \mathbf{R}_i\}_{i>0}$  by optimizing the loss:

$$\mathcal{L}' = w_o \mathcal{L}_o\left(\mathcal{P}_0 \bigcup_{j=1}^{j<i} \mathcal{P}'_j, \mathcal{P}_i\right) + w_v \mathcal{L}_v + w_r \mathcal{L}_r, \quad (14)$$

where  $\mathcal{P}'_j$  is the predicted point cloud at optimized waypoint  $(\mathbf{p}'_j, \mathbf{R}'_j)$ . During the optimization, we downsample

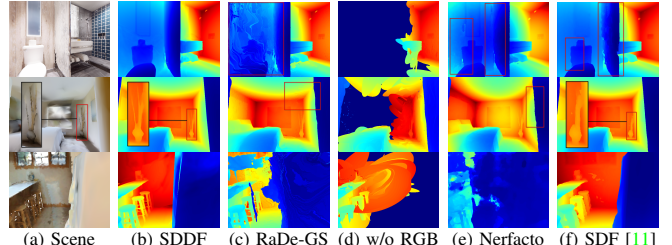


Fig. 4: Qualitative comparison of SDDF predictions. Row 1: Replica Hotel (synthesized). Row 2: Gibson Allensville (synthesized). Row 3: ScanNet scene 0000-00 (real). In areas with limited sensor measurements, RaDe-GS [12] fails to learn the geometry, with RGB (c) or without RGB (d), yielding artifacts. Nerfacto [13] in (e) shows significant artifacts and large distance prediction error. SDF-Instant-NGP [11] in (f) tends to learn a smoother approximation, missing sharp boundaries.

$\mathcal{P}_0 \bigcup_{j=1}^{j<i} \mathcal{P}'_j$  with a stride of  $i - j$ , labeled as  $\tilde{\mathcal{P}}_j$ , such that  $\tilde{\mathcal{P}}_j$  has a constant size. This incremental optimization strategy provides two benefits. First, it reduces the use of GPU memory and makes the along-trajectory multi-view optimization problem solvable. Second, it allows the robot to parallelize the trajectory optimization and execution.

### IV. EVALUATION

**SDDF Reconstruction Experiments.** We convert the sensor poses and range measurements  $\{\mathbf{T}_t, \mathcal{Z}_t\}_{t=1}^T$ ,  $\mathcal{Z}_t = \{\mathbf{v}_i, r_{t,i}\}_{i=1}^N$ , described in Sec. II, into a dataset  $\mathcal{D} = \{\mathbf{p}_j, \mathbf{v}_j, f_j^*, i_j^*, s_j^*\}_j$  suitable for training our SDDF model. A total of 14 synthesized (LiDAR and depth camera) datasets are used for comparison. We obtained data from six scenes from Replica (“Hotel” and “Office 0-5”) [14] and the Allensville scene from Gibson [15].

We compare our method against three baselines: Nerfacto [13], RaDe-GS [12], and SDF-Instant-NGP [11]. These methods were not initially designed for SDDF prediction but can be used to predict SDDF as follows. For SDF, we implement sphere tracing [16] to find the closest point on the surface along the query direction. For Nerfacto and RaDe-GS, we render a depth image at the query viewpoint, project the depth image to a point cloud, and compute the distance.

Fig. 4 shows qualitative comparisons against RaDe-GS [12] (RGB-D and depth-only), Nerfacto [13], and SDF-Instant-NGP [11] with sphere tracing. In Fig. 4c, RaDe-GS exhibits erroneous artifacts around the toilet, at the corner, or near the fridge, where there are limited RGB-D observations. Meanwhile, our method accurately reconstructs these areas. Nerfacto also exhibits artifacts due to insufficient data as shown in Fig. 4e. Since no explicit representations like our ellipsoids or the Gaussians in RaDe-GS are used, Nerfacto predicts SDDF based on the learned volume density, which is optimized for photometric rendering rather than scene geometry, leading to large distance prediction errors.

Meanwhile, sphere tracing on SDF-Instant-NGP [11] does not exhibit significant artifacts. However, this baseline tends to learn smoother shapes that lack sharper details, such as the plant shown in the second row or the chairs in the third row of Fig. 4f. Moreover, the first row of Fig. 4f shows that the errors accumulated during sphere tracing become significant at boundaries when the SDF model does not predict accurate

