From Distances to Trajectories: Real-Time Signed Distance Function Mapping and Distance-Accelerated Motion Planning for UAVs

Jason Stanley[1*] Zhirui Dai[1*] Tzu-Chin Ho[1] Siddharth Saha[2] Christopher Barngrover[2] Nikolay Atanasov[1]

***Abstract*— Autonomous aerial inspection of maritime infrastructure, such as large vessels and offshore assets, enables safer and more efficient operation in hazardous environments like confined ballast tanks. These settings involve complex, partially unknown geometry with tight clearances and limited visibility, posing significant challenges for real-time UAV navigation under strict onboard computational constraints. We present a unified framework for efficient onboard signed distance function (SDF) reconstruction and motion planning in unknown environments. Our approach reconstructs a non-truncated Euclidean SDF in real time by combining an explicit octree prior with an implicit neural residual, enabling accurate and differentiable distance estimates from streaming depth data. We further introduce Bubble⋆, a search-based planner that expands collision-free regions derived from SDF values and efficiently generates safe corridors for dynamically feasible trajectory optimization in confined environments. We demonstrate real-time, fully onboard autonomous flight in previously unseen environments, highlighting the effectiveness of tightly coupled SDF-based mapping and planning for UAV inspection of maritime assets.**

I. Introduction

Autonomous aerial inspection of maritime infrastructure, including large vessels and offshore assets, offers a safer and more efficient alternative to human inspection in hazardous environments. These settings involve complex, partially unknown geometry, tight clearances, and limited visibility, making reliable navigation particularly challenging for UAVs operating with constrained onboard sensing and computation.

Reliable navigation in such environments depends critically on accurate geometric representations. Signed distance functions (SDFs) provide a continuous encoding of scene geometry that supports both the construction of safe volumetric regions (*bubbles*) for motion planning and the extraction of surface meshes for structural inspection and digital reconstruction. However, existing SDF reconstruction methods either require substantial computational resources or sacrifice global accuracy, limiting their use onboard UAVs.

To address these challenges, we propose **O**ctree **Re**sidual **N**etwork (OREN), an efficient method for online reconstruction of a non-truncated Euclidean SDF from streaming depth observations. Our approach combines an explicit octree prior with an implicit neural residual, producing globally consistent and differentiable distance estimates while maintaining real-time performance.

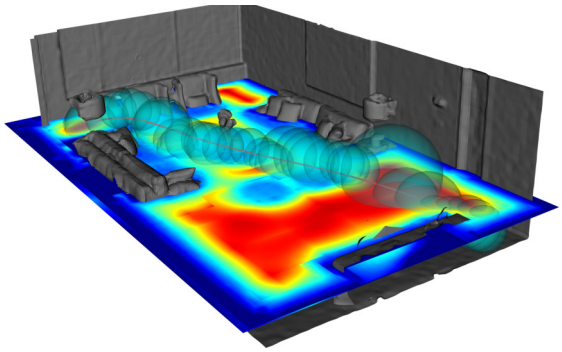


Fig. 1: An octree residual network (OREN) reconstructs accurate Euclidean signed distance function online, which is used by Bubble⋆ to plan a safe flight corridor. The color map indicates the distance to the environment surfaces (large/red to small/blue), while the cyan spheres show the planned sequence of safe bubbles.

This SDF representation enables Bubble⋆, a search-based planner that expands collision-free bubbles. Inspired by Jump Point Search (JPS) [1], Bubble⋆ leverages the SDF to perform clearance-driven expansions, enabling efficient exploration of free space. The resulting overlapping bubbles define safe corridors that are directly usable for trajectory optimization (see Fig. 1).

In summary, this work makes the following contributions:

- We develop OREN, an explicit-implicit model for fast, globally consistent Euclidean SDF reconstruction.
- Bubble⋆, an efficient motion planning algorithm that exploits SDF structure to generate safe corridors.
- Real-world validation of fully onboard autonomous UAV navigation in previously unseen environments.

II. Preliminaries

We consider a quadrotor robot with state

$$x = (p, v, R, \omega) \in \mathcal{X} := \mathbb{R}^3 \times \mathbb{R}^3 \times SO(3) \times \mathbb{R}^3, \quad (1)$$

where p and v are position and velocity in the inertial frame, $R \in SO(3)$ is orientation, and ω is body angular velocity. The control input is $u = (F, \tau) \in \mathbb{R} \times \mathbb{R}^3$, consisting of collective thrust F and body torque τ. The quadrotor dynamics are modeled as:

$$\dot{x} = f(x) + G(x)u = \begin{cases} \dot{p} = v, \\ \dot{v} = -ge_3 + \frac{1}{m} RFe_3, \\ \dot{R} = R\hat{\omega}, \\ \dot{\omega} = J^{-1}(\tau - \hat{\omega}J\omega), \end{cases} \quad (2)$$

*Equal contribution

[1]The authors are with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA (e-mails: {jtstanle,zhdai,tzh005,natanasov}@ucsd.edu).

[2] The authors are with Shield AI, 600 W Broadway Suite #600, San Diego, CA, 92101, USA.

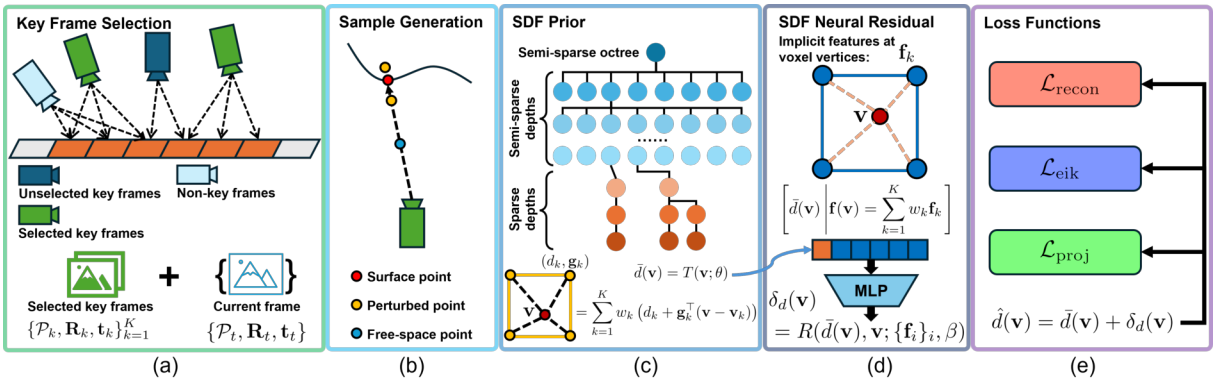


Fig. 2: Overview of OREN: (a) keyframe selection maximizing coverage with minimal overlap; (b) sampling of surface, near-surface, and free-space points; (c) SDF prior $\bar{d}(\mathbf{v})$ from gradient-augmented octree interpolation; (d) neural feature interpolation and MLP prediction of residual $\delta_d(\mathbf{v})$; (e) final SDF $\hat{d}(\mathbf{v}) = \bar{d}(\mathbf{v}) + \delta_d(\mathbf{v})$, trained with reconstruction, Eikonal, and projection losses.

where m is the mass, J is the inertia, $e_3 = (0,0,1)^\top$, g is the gravity magnitude, and $\hat{\omega}$ is a skew-symmetric matrix obtained from $\omega \in \mathbb{R}^3$.

The system (2) is differentially flat. A system is said to be differentially flat if there exists an output z such that the state x and input u can be expressed as algebraic functions of z and a finite number of its derivatives:

$$x = \Psi_x(z, \dot{z}, \ldots, z^{(s-1)}), \quad u = \Psi_u(z, \dot{z}, \ldots, z^{(s)}), \tag{3}$$

for some integer s, where the mappings Ψ_x and Ψ_u are known in closed form for quadrotors [2], [3]. For a quadrotor, flat outputs $z = (p, \psi)$, where ψ is yaw, with $s = 3$ enable trajectory generation via piecewise polynomials, with dynamic feasibility imposed on derivatives of $z(t)$ and obstacle avoidance on $p(t)$.

We consider a quadrotor operating in an unknown environment with obstacles denoted by $\mathcal{O} \subset \mathbb{R}^3$. The signed distance function (SDF) $d : \mathbb{R}^3 \to \mathbb{R}$ is defined as

$$d(p) = \begin{cases} \min_{q \in \partial\mathcal{O}} \|p - q\|_2, & p \notin \mathcal{O}, \\ -\min_{q \in \partial\mathcal{O}} \|p - q\|_2, & p \in \mathcal{O}. \end{cases} \tag{4}$$

The SDF satisfies $d(p) > 0$ in free space, $d(p) = 0$ on boundaries, and directly encodes clearance to the nearest obstacle surface.

III. Problem Statement

Given a quadrotor with dynamics (2) and an onboard range sensor, we reconstruct an SDF from point clouds and use it to plan dynamically feasible collision-free trajectories. Given start and goal states $x_s, x_g \in \mathcal{X}$, we seek a trajectory $(x(t), u(t))$ and final time T minimizing control effort subject to:

$$\begin{aligned} \min_{x(\cdot),u(\cdot),T} \quad & \int_0^T u(t)^\top \mathbf{W} u(t)\, dt + \rho(T) \\ \text{s.t.} \quad & \dot{x}(t) = f(x(t)) + G(x(t))u(t), \\ & p(t) \notin \mathcal{O}, \quad \forall t \in [0, T], \\ & x(0) = x_s, \quad x(T) = x_g, \end{aligned} \tag{5}$$

Assumption 1 (ε-Clear Path)**.** *There exists a feasible trajectory $x^\star(t)$ connecting x_s to x_g whose position satisfies*

$$d(p^\star(t)) \geq \varepsilon, \quad \forall t \in [0, T^\star], \exists\, \varepsilon > 0 \tag{6}$$

This assumption ensures the existence of a collision-free trajectory with nonzero clearance, excluding degenerate solutions that graze obstacle boundaries and enabling the construction of safe free-space regions.

In the following sections, we present a method that jointly constructs an SDF from range measurements and exploits the SDF structure to enable efficient trajectory optimization under the constraints in (5).

IV. SDF Reconstruction with OREN

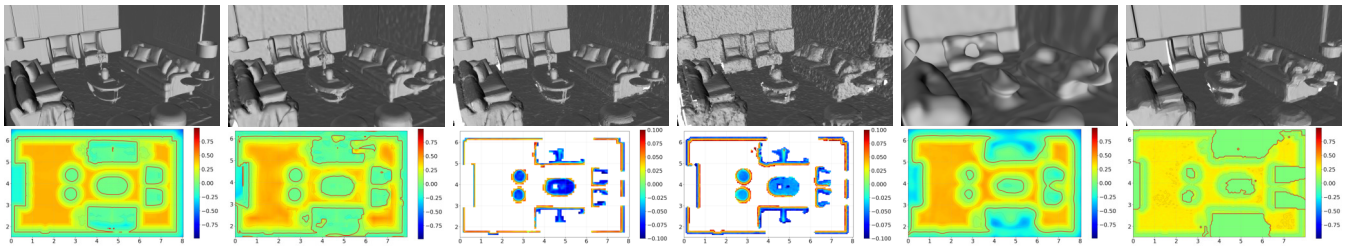
To solve the problem, we first propose OREN, as shown in Fig. 2, which uses a hybrid representation that combines an explicit SDF prior from gradient-augmented octree interpolation with a neural residual correction. With point clouds converted from the sensor measurements, OREN is trained online by actively selecting key frames, generating training samples and using loss terms to enforce geometric accuracy and SDF properties.

A. Octree Data Structure

As shown in Fig. 2, we construct a semi-sparse octree to cover the space. For a leaf octant, each vertex $\mathbf{v}_k$ stores the following learnable parameters: an SDF estimate d_k, the corresponding SDF gradient $\mathbf{g}_k$ and an implicit neural feature $\mathbf{f}_k$. The SDF estimate and gradient are used to compute an SDF prior at query points inside the octant via interpolation, while the implicit feature is used to predict a residual correction to the interpolation via a neural network decoder.

B. Prior for SDF Reconstruction

For a query point $\mathbf{v}$, we locate the smallest containing octant and estimate the SDF by gradient-augmented interpolation of the priors stored at the eight corner vertices. First, each vertex yields the corresponding extrapolated value,



(a) Ground Truth (b) Ours (c) H_2-Mapping [4] (d) PIN-SLAM [5] (e) HIO-SDF [6] (f) Voxblox [7]

Fig. 3: Qualitative comparison of mesh reconstruction (top row) and z-plane slice of SDF reconstruction (bottom row) on Replica room 0 [8]. OREN reconstructs a mesh with the highest completion ratio and accurate SDF both near and far from the surface. H_2-Mapping and PIN-SLAM only learn truncated SDF. HIO-SDF learns an over smooth result. Voxblox significantly under-estimates the SDF.

$d_k(\mathbf{v}) = d_k + \mathbf{g}_k^\top(\mathbf{v} - \mathbf{v}_k)$. Then, the prior SDF is obtained by weighted interpolation of these extrapolations:

$$\bar{d}(\mathbf{v}) = \frac{1}{\gamma}\sum_{k=1}^{8} w_k d_k(\mathbf{v}), \tag{7}$$

where w_k is the standard trilinear interpolation weight and $\gamma = \sum_k w_k$. Compared to standard trilinear interpolation, incorporating gradient information improves accuracy and stability, particularly near surfaces.

C. Neural Residual for SDF Reconstruction

The accuracy of the SDF prior is limited to the octree resolution, causing lack of geometric details. To increase the fidelity, we learn a residual correction $\delta_d(\mathbf{v})$ to the interpolated distance $\bar{d}(\mathbf{v})$ in (7) via a neural network.

First, we obtain the tri-linearly interpolated implicit feature at query point $\mathbf{v}$: $\mathbf{f}(\mathbf{v}) = \sum_k w_k \mathbf{f}_k$ with the eight vertex implicit features. Then, a small MLP decoder D predicts a residual distance with the explicit SDF prior $\bar{d}(\mathbf{v})$ and the implicit neural feature $\mathbf{f}(\mathbf{v})$: $\delta_d(\mathbf{v}) = D(\bar{d}(\mathbf{v}), \mathbf{f}(\mathbf{v}); \beta)$, such that the final SDF estimate is $\hat{d}(\mathbf{v}) = \bar{d}(\mathbf{v}) + \delta_d(\mathbf{v})$. This hybrid formulation combines the efficiency and global consistency of the explicit prior with the expressiveness of the implicit neural residual.

D. Key Frame Selection, Sampling, and Training

To maintain real-time performance, training uses keyframes selected by viewpoint novelty, with a fixed-size window maximizing scene coverage. Range measurements from these frames generate three types of training samples: free-space points along sensor rays, surface points at ray endpoints, and near-surface perturbations. These samples provide supervision across both free space and surfaces.

OREN is trained with losses enforcing geometric accuracy and SDF properties. A reconstruction loss supervises surface and near-surface distances, an Eikonal loss for unit-norm gradients, and a projection loss provides additional supervision in free space. Together, these objectives produce a continuous SDF that is accurate near observed surfaces while maintaining valid distance estimates in free space.

V. SDF-Based Motion Planning with Bubble*

Given an SDF (Sec. II) and start and goal positions, we seek a collision-free path. Grid-based search (e.g., A*) expands neighboring cells to explore free space but does not exploit SDF geometry. We propose Bubble*, a planner that leverages OREN to expand collision-free regions. Like A*, it prioritizes nodes using a heuristic where $h(p) = \|p - p_g\|_2$ is the Euclidean distance to the goal. Using OREN, the planner grows *bubbles* that span multiple cells, enabling long-range expansions and increasing efficiency. Since any two points within a bubble are mutually collision-free, this permits off-grid connections without cell-by-cell propagation.

Algorithm 1: Bubble* Planner

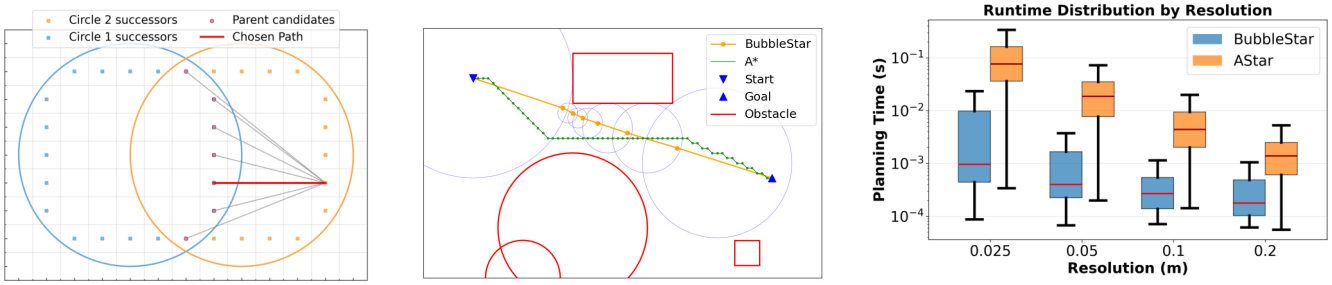
Input: p_s, p_g, SDF $d(\cdot)$, heuristic $h(\cdot)$
Output: Path and Bubble corridor from p_s to p_g
Initialize OPEN $\leftarrow \{p_s\}$, CLOSED $\leftarrow \emptyset$
Initialize $g(p_s) = 0$ and $g(p) = \infty$ for all $p \neq p_s$
while *OPEN* $\neq \emptyset$ **do**
  $c \leftarrow \arg\min_{p \in \text{OPEN}} \big(g(p) + h(p)\big)$
  OPEN $\leftarrow$ OPEN $\setminus \{c\}$
  $\mathcal{B}_c \leftarrow \mathcal{B}(c, d(c))$
  if $p_g = c$ **then**
    return Backtrack(p_g)
  $S \leftarrow S(\mathcal{B}_c) \setminus$ CLOSED
  CalculateSuccessors$(S, \mathcal{B}_c, \text{OPEN}, \text{Parent}(\mathcal{B}_c))$
  CLOSED $\leftarrow$ CLOSED $\cup \big(\mathcal{B}_c \setminus S(\mathcal{B}_c)\big)$
return failure

A. Bubble Representation and Search

We describe the Bubble* algorithm, which operates over collision-free regions induced by the SDF. Let $\mathcal{V}^{\text{free}}$ denote the set of free grid centers. For each $c \in \mathcal{V}^{\text{free}}$, define a bubble $\mathcal{B}_c := \mathcal{B}(c, \hat{d}(c))$, where $\hat{d}(c)$ is the SDF estimate produced by OREN at grid center c. Bubble* generates successors on the boundary of the current bubble rather than expanding neighboring grid cells. Let $\mathcal{N}_{2d}(p)$ denote the axis-aligned neighbors of a grid point p. The successor set is

$$S(\mathcal{B}_c) := \{p \in \mathcal{B}_c \mid \exists q \in \mathcal{N}_{2d}(p) \text{ s.t. } q \notin \mathcal{B}_c\}. \tag{8}$$

Each $p \in S(\mathcal{B}_c)$ defines a new candidate bubble center. The procedure CalculateSuccessors (Alg. 1) updates their *cost-to-come* $g(p)$ by connecting to previously discovered nodes within $\mathcal{B}_c$ (Fig. 4a). Interior points of $\mathcal{B}_c$ are marked as explored to avoid redundant expansion. The search terminates



(a) (b) (c)

Fig. 4: (a) Successors select a parent in the current bubble according to the minimum *cost-to-come*; existing (blue) and new (orange) nodes together form the open list. (b) Representative Bubble* vs. A* comparison in a 2D environment; $h(p) = \|p - p_g\|_2$ is used as the heuristic. (c) We evaluate the run-time of both Bubble* and A* on 1,000 randomly selected start and goal pairs at various grid resolutions in the 2D environment shown in (b). The A* planner uses the sign of the SDF to determine occupancy. Bubble* achieves shorter or equal path length in every case.

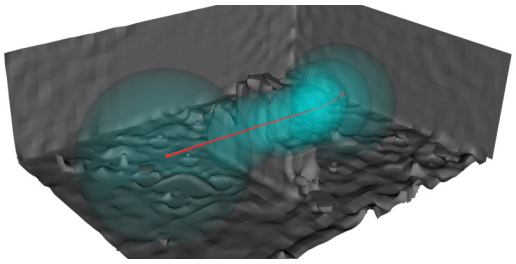


Fig. 5: SDF Mesh (gray), bubble corridor (cyan), and trajectory (red) generated onboard a UAV in real-time.

when the goal lies within the current bubble, after which the path is recovered via parent pointers. Under the ε-clear path assumption (Assumption 1), Bubble* is complete and optimal with respect to the underlying grid.

B. Trajectory Optimization

Bubble* also recovers an ordered sequence of bubbles $\boldsymbol{\mathcal{B}} = \{\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_M\}$, whose union defines a continuous collision-free corridor from p_s to p_g to serve as the feasible region for trajectory generation. We then compute a dynamically feasible trajectory constrained to the corridor, $p(t) \in \bigcup_{i=0}^{M} \mathcal{B}_i \ \forall t \in [0, T]$, approximating (5). Waypoints are constrained to the overlap of consecutive bubbles, ensuring collision-free motion while preserving optimization flexibility. Since these overlaps lie in free space, the resulting trajectory satisfies the requirement under Assumption 1.

Trajectory generation follows the differential flatness formulation in Sec. II and uses the MINCO representation [3]. The trajectory is parameterized by spatial waypoints and segment durations, defining a piecewise-polynomial trajectory satisfying (2). Gradients from MINCO enable efficient optimization of control effort and flight time, and the trajectory is mapped to state-input space via flatness to yield a dynamically feasible motion from start to goal. The compact bubble corridor from Bubble* directly reduces the complexity of trajectory generation compared to planning over raw occupancy grids.

VI. Results

We evaluate the proposed system, which tightly integrates real-time SDF reconstruction, Bubble* planning, and trajectory optimization to enable efficient, fully onboard UAV flight in unknown environments.

Fig. 3 shows results on the Replica dataset [8]. H_2-Mapping, PIN-SLAM, and OREN produce high-quality meshes. H_2-Mapping yields smoother surfaces due to surface focused voxel allocation, while our method achieves more complete reconstructions. Compared to PIN-SLAM [5], which predicts SDF values primarily near surfaces, our method produces smoother geometry with less noise. Relative to HIO-SDF [6], our reconstructions exhibit higher fidelity across the workspace, demonstrating accurate, globally consistent Euclidean SDF estimates suitable for planning.

We next evaluate planning efficiency. Fig. 4c compares Bubble* with A* in randomized 2D environments. By leveraging SDF-based clearance to expand collision-free regions, Bubble* reduces node expansions and collision checks, achieving 91–99% fewer collision checks across resolutions while producing shorter paths.

Real-world experiments on a quadrotor equipped with a Jetson Orin NX demonstrate the full system running fully onboard at *7 Hz*, enabling continuous mapping and planning in previously unseen environments (Fig. 5).

These results demonstrate that real-time Euclidean SDF reconstruction combined with bubble planning enables efficient, safe, and fully onboard autonomous UAV navigation.

VII. Conclusion

We presented a unified approach for SDF reconstruction and SDF-based motion planning and demonstrated that it enables autonomous UAV navigation in unknown environments. We achieved efficient, globally consistent Euclidean SDF estimation from streaming point-cloud data using an octree residual network, OREN, and formulated a search-based planning algorithm, Bubble*, that exploits the SDF to generate safe corridors for trajectory optimization. Our experiments demonstrated improved reconstruction quality, planning efficiency, and reliable autonomous flight in previously unseen environments. These results are particularly relevant for aerial inspection of maritime infrastructure, where confined, partially unknown geometries challenge conventional approaches and where fully onboard, efficient navigation is essential.

REFERENCES

[1] D. Harabor and A. Grastien, "Online Graph Pruning for Pathfinding On Grid Maps," *AAAI Conference on Artificial Intelligence*, 2011.

[2] D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors," in *IEEE International Conference on Robotics and Automation*, 2011.

[3] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically Constrained Trajectory Optimization for Multicopters," *IEEE Transactions on Robotics*, 2022.

[4] C. Jiang, H. Zhang, P. Liu, Z. Yu, H. Cheng, B. Zhou, and S. Shen, "H2-Mapping: Real-time Dense Mapping Using Hierarchical Hybrid Representation," *IEEE Robotics and Automation Letters*, 2023.

[5] Y. Pan, X. Zhong, L. Wiesmann, T. Posewsky, J. Behley, and C. Stachniss, "PIN-SLAM: LiDAR SLAM Using a Point-Based Implicit Neural Representation for Achieving Global Map Consistency," *IEEE Transactions on Robotics*, 2024.

[6] V. Vasilopoulos, S. Garg, J. Huh, B. Lee, and V. Isler, "HIO-SDF: Hierarchical Incremental Online Signed Distance Fields," 2024.

[7] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-board MAV planning," 2017.

[8] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe, "The Replica Dataset: A Digital Replica of Indoor Spaces," *arXiv preprint arXiv:1906.05797*, 2019.