Manifold Embedding for Fast and Accurate 3D Reconstruction

Duo Chen, Zixin Tang, Ke Song, Xingyu Peng, Wuque Cai, Hongze Sun, Dezhong Yao, *Senior Member, IEEE*, and Daqing Guo

***Abstract*—The goal of the fusion process in RGB-D reconstruction systems is to verify and update the 3D model while ensuring both completeness and accuracy. However, achieving precise dense correspondences in a point-to-point or pixel model during this process is challenging and computationally intensive. To address this challenge, we propose a Manifold Embedding framework that facilitates rapid point-to-surface fusion, removing the need for direct point-to-point or pixel correspondences. Our approach consists of three main steps: 1) *Manifold Voxel:* We transform discrete point sets into smooth surfaces using the Implicit Moving Least Squares (IMLS) method; 2) *Two-Step Filtering:* We enhance reconstruction accuracy through a two-step filtering technique that evaluates sampling points based on probabilistic measures; 3) *Embedding for Smooth Surface:* Lastly, we embed points into a smooth manifold surface represented via IMLS, ensuring high-quality reconstructed surfaces. Extensive experiments on both real and synthetic 3D scenes demonstrate the effectiveness of our Manifold Embedding framework. For instance, on the public *Replica* dataset, our method surpasses state-of-the-art fusion techniques regarding both completeness and accuracy. Our average accuracy is 2.11 cm and completeness is 2.80 cm, while NICE-SLAM achieves 2.85 cm and 3.00 cm, respectively (with lower values indicating better performance). Overall, our proposed method provides superior reconstruction quality and enhanced computational efficiency (See Fig. 1).**

***Index Terms*—3D reconstruction, fusion, implicit moving least squares, manifold.**

Received 25 July 2024; revised 16 December 2024; accepted 25 January 2025. Date of publication 21 July 2025; date of current version 21 October 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFF1204200, in part by the Natural Science Foundation of Chongqing, China under Grant CSTB2024NSCQ-MSX0627, in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZD-K202401603, in part by China Postdoctoral Science Foundation under Grant 2024M763876, in part by the High Level Talent Research Initiation Project of Chongqing University of Education under Grant 2023BSRC019, and in part by the Youth Project of Sichuan Natural Science Foundation of China under Grant 2024NSFSC1452. The associate editor coordinating the review of this article and approving it for publication was Prof. Sanghoon Lee. *(Corresponding authors: Zixin Tang; Daqing Guo.)*

Duo Chen is with the School of Artificial Intelligence, Chongqing University of Education, Chongqing 400065, China, also with the MOE Key Lab for NeuroInformation, School of Life Science and Technology, Clinical Hospital of Chengdu Brain Science Institute, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with Chongqing University Industrial Technology Research Institute, Chongqing 401331, China.

Zixin Tang is with the School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu 611130, China (e-mail: tangzx@swufe.edu.cn).

Ke Song and Xingyu Peng are with the School of Artificial Intelligence, Chongqing University of Education, Chongqing 400065, China.

Wuque Cai, Hongze Sun, Dezhong Yao, and Daqing Guo are with the MOE Key Lab for NeuroInformation, School of Life Science and Technology, Clinical Hospital of Chengdu Brain Science Institute, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: dqguo@uestc.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMM.2025.3590908, provided by the authors.

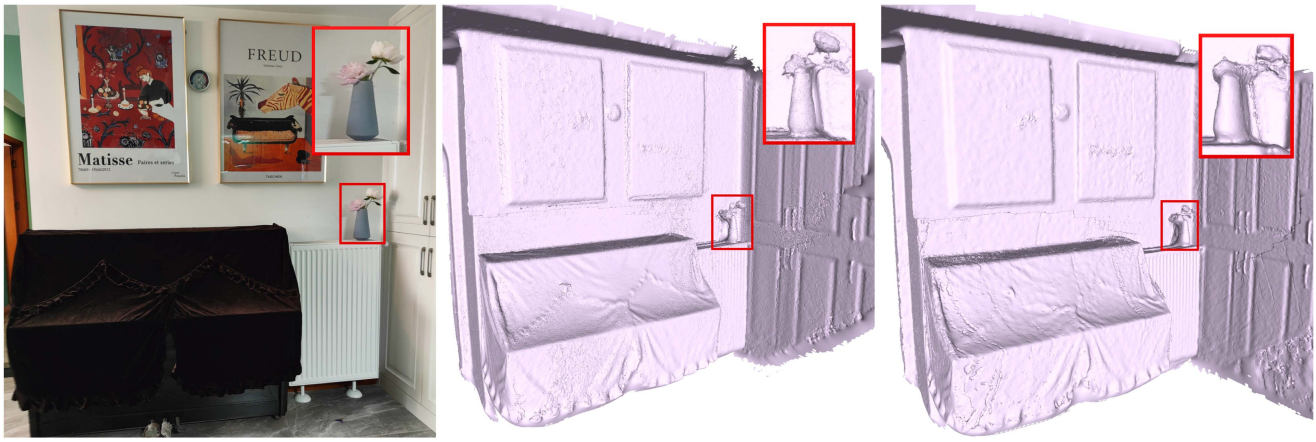
Digital Object Identifier 10.1109/TMM.2025.3590908

I. Introduction

THE reconstruction or fusion of 3D data requires precise and dense correspondence to effectively verify and update the accumulated 3D model, ensuring both completeness and accuracy. This dense correspondence is usually achieved through point-to-pixel methods that use the projective model to map 3D points onto a depth map located within a pixel grid. By utilizing the inverse global camera pose and intrinsic parameters, each point in the global model can be projected onto the image plane of the current RGB-D camera view (depth map), establishing point-to-pixel correspondence [2]. This technique has been widely demonstrated in the research conducted by Keller et al. [2], Whelan et al. [3], and Schöps et al. [4]. However, it is important to note that this projection process can be affected by disturbances due to uncertainties in depth and pose estimations.

Depth Uncertainties primarily arise from sensor noise, which can vary significantly between different imaging systems. For example, stereo sensors often face matching errors, while LiDAR sensors are affected by factors such as ambient light and reflective rates. This variability makes it challenging to train a unified network that can effectively handle the diverse characteristics of these imaging modalities. In this study, we focus on Time-of-Flight based sensors, specifically the Intel RealSense™ L515 and the Microsoft Azure Kinect™ DK, both of which are well-established in the field of 3D reconstruction via RGB-D sensors. The depth root mean squared error (RMSE) of the Intel RealSense™ is reported to be 5 mm at a distance of 1 m and 14 mm at a distance of 9 m. It also has a claimed standard deviation of 2.5 mm and 15.5 mm for ideal targets with 95% reflectivity (under ambient light of less than 500 lux), according to Intel. In our experiments conducted with real-world indoor scenes, we observed depth errors that often reached several centimeters—typically around $3-5$ cm at 5 m when scanning a white wall in low ambient light conditions. The depth RMSE of the Microsoft Azure Kinect™ DK sensor is comparable to that of the Intel RealSense™, showcasing similar errors in indoor scans.

Pose Uncertainties arise during the pose estimation process, typically measured by assessing camera tracking accuracy



(a) Ground Truth (b) **Manifold Embedding** (c) Cyclical Fusion [1]

Fig. 1. *Comparison of Manifold Embedding and Cyclical Fusion in Real-World Scanning.* In our study, we demonstrate that our method surpasses Cyclical Fusion [1] in both processing speed (50 ms compared to 130 ms) and accuracy, particularly in the preservation of intricate details.

through Absolute Trajectory Error (ATE) [5]. To compute the ATE, the evaluated trajectory is first aligned rigidly with the ground truth. After this alignment, the ATE is calculated as the average of pose discrepancies across all frames [6]. The advanced pose estimation system ROSEFusion has achieved an average ATE of 2.6 cm on the FastCaMo-Synth dataset [6].

These uncertainties can lead to incorrect correspondences, resulting in incomplete or inaccurate 3D models. Recent research has addressed these challenges by introducing non-projective models based on optimal transport theory [7] to improve reconstruction accuracy. These models establish dense correspondences between points by minimizing the L^2-Wasserstein distance. While these approaches enhance accuracy, they require substantial computational resources due to the large number of potential point-to-point correspondences involved in solving the optimal transport problem. Therefore, achieving a balance between accuracy and efficiency remains a significant challenge, particularly for mobile devices that face limitations in battery life.

Learning Methods have received significant attention in the field of 3D reconstruction [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. However, these learning-based approaches often show limited robustness when dealing with sensor noise, especially when the imaging systems used differ from those represented in the training datasets. Several factors constrain the feasibility of training a neural network for 3D reconstruction. Firstly, consumer-level sensors, particularly when combined with battery-powered devices, typically have limited computational resources. This makes them unsuitable for resource-intensive tasks, such as training deep learning models. For instance, DI-Fusion [13] operates at a processing speed of only 10 Hz on modern GPU platforms, and its training can take several days or even weeks. Additionally, other end-to-end networks, like NICE-SLAM [11], rely on a backpropagation framework to update the 3D representation. These networks require advanced GPUs (RTX3090 TDP350 W) with 24 GB of memory to effectively reconstruct indoor scenes. Moreover, learning methodologies are heavily dependent on specific datasets, which are often synthetic and may not accurately reflect real-world scanning conditions. For example, IMLSnet [15] is trained using the ShapeNet dataset but faces significant challenges when processing actual scan inputs. In terms of depth uncertainties, the diversity of imaging systems leads to various noise characteristics. For instance, LiDAR sensors can be influenced by ambient light and surface reflective properties, while stereo sensors often encounter matching errors. Even within the LiDAR category, there is substantial variability between solid-state and mechanical LiDARs. As a result, the challenge of training a unified network to accommodate these diverse imaging systems remains unresolved. Therefore, we have chosen to adopt a classical approach, leveraging mathematical tools to design an *explainable algorithm* that effectively addresses this challenge.

Critically, achieving point-to-point or pixel dense correspondence is not always necessary for successful data fusion. Instead, a point-to-surface fusion paradigm that employs robust local geometrical constraints can be more effective, particularly when processing noisy input data. In this context, our paper introduces an innovative framework called Manifold Embedding, which utilizes a point-to-surface fusion approach. This method enables the integration of incoming points into a smooth manifold represented by the implicit moving least-squares (IMLS) surface. To achieve this, we aim to accurately characterize the local surface by using a set of sample points through the moving least-squares technique. Our proposed fusion framework breaks the fusion process into two distinct filters: Verification and Updating Probability Filtering. These filters effectively address uncertainties that may interfere with the reconstruction process. The overall methodology of our proposed approach is illustrated in Fig. 2 and primarily contributes to the field in the following ways:

- We propose a point-to-surface fusion method that eliminates the complex and time-consuming task of estimating point-to-point or pixel correspondences, allowing the

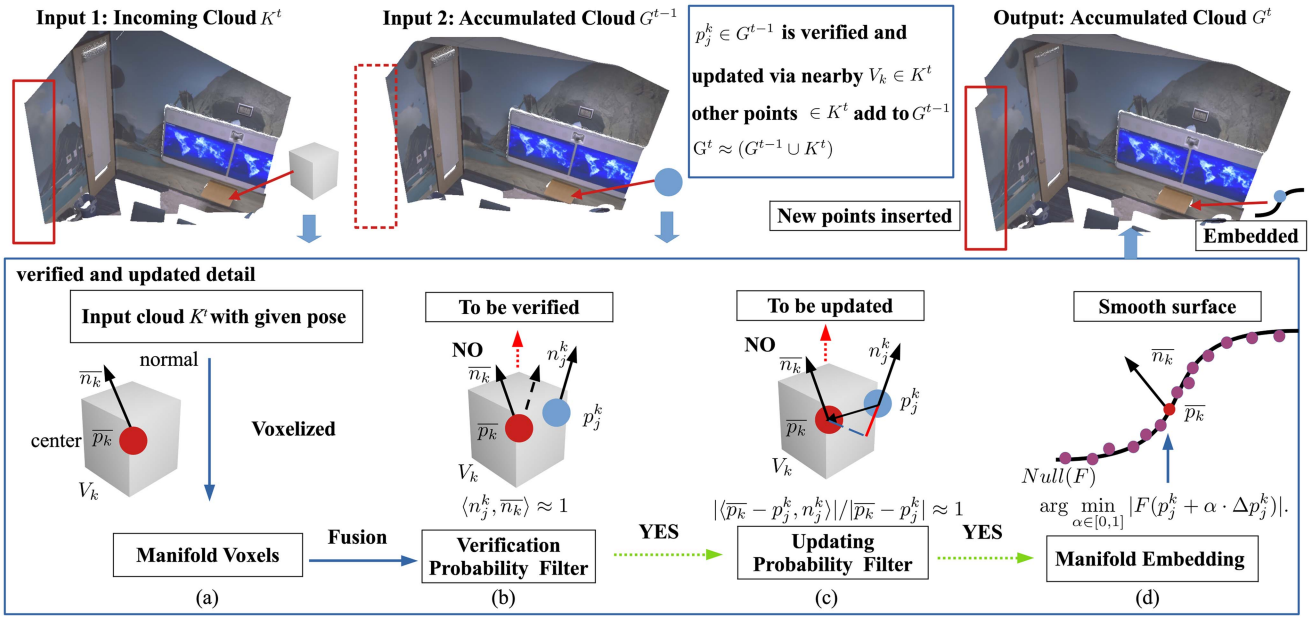


Fig. 2. The manifold embedding process integrates voxels from the incoming cloud K^t into the accumulated cloud G^{t-1} using two-step probability filters and an Implicit Moving Least Squares (IMLS) function F, resulting in the smooth embedded cloud G^t. The process involves four steps: (a) **Generating Manifold Voxels** (V_k): Voxelization of K^t creates manifold voxels characterized by center points $\overline{p_k}$ and normal vectors $\overline{n_k}$. (b) **Verification Probability Filter**: Each voxel V_k is validated against adjacent points p_j^k in G^{t-1}, ensuring normal alignment with $\langle n_j^k, \overline{n_k} \rangle \approx 1$. (c) **Updating Probability Filters**: Verified points satisfy the proximity condition $|\langle \overline{p_k} - p_j^k, n_j^k \rangle| / |\overline{p_k} - p_j^k| \approx 1$, ensuring spatial coherence. (d) **Manifold Embedding**: A smooth surface is constructed by minimizing $|F(p_j^k + \alpha \Delta p_j^k)|$, ensuring proper embedding of points p_j^k.

fusion process to operate without the constraints of dense point or pixel relationships.

- The fusion technique is guided by an implicit moving least-squares surface function, which takes full advantage of the geometric properties of the 3D surface, effectively reducing the impact of uncertainty.
- The proposed fusion framework decouples the process into two distinct filters: Verification and Updating Probability Filtering. This novel approach replaces fixed thresholds with probabilistic measures, enhancing the robustness of the fusion process. Extensive experimental results support the effectiveness of the proposed framework.

II. Related Work

Reconstruction with high geometric quality is a vital task in the fields of computer vision, graphics, and multimedia, as it greatly enhances various downstream applications, including object detection [19], [20] and re-identification [21]. This reconstruction process primarily utilizes projective and non-projective models, which are characterized by different geometric representations, such as voxel-based methods and surfel/point-based methods.

Voxel-based Methods [6], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33] have made significant strides in 3D reconstruction, particularly through the efforts of Newcombe et al. [22], Niessner et al. [23], Zhou et al. [24], [25], and Dai et al. [26]. These methods employ volumetric data and represent surface geometry using the truncated signed distance function (TSDF) in voxel grids. A common strategy for establishing voxel correspondence is to perform a nearest neighbor lookup on depth maps, which facilitates projective TSDF measurement. This is often followed by surface updates using a weighted average function.

In addition to voxel methods, various learning-based approaches have been developed to enhance reconstruction quality. Notable techniques include NICE-SLAM [11], iMap [12], DI-Fusion [13], NeuralFusion [14], and IMLSNet [15]. For instance, NICE-SLAM leverages hierarchical scene representations to capture multi-level local information, improving reconstruction accuracy in expansive indoor spaces by utilizing pre-trained geometric priors. Similarly, iMap features an innovative keyframe structure combined with a multi-processing framework that uses dynamic information-guided pixel sampling to increase computational efficiency. IMLSNet utilizes an octree structure to generate Moving Least-Squares (MLS) points as needed, effectively capturing shape geometry with locally learned priors.

Despite their advantages, these end-to-end networks often require substantial GPU memory (typically around 24 GB) to maintain low mapping and fusion rates (ranging from 2 to 10 Hz). They are also especially sensitive to sensor noise, particularly when the noise diverges from the training data. In contrast, the proposed method tackles these issues by introducing a verification-updating approach for manifold embedding that operates independently of prior information. This innovation enables robust 3D reconstruction capabilities, even in the presence of sensor noise or differences across various imaging systems.

Surfel and Point Representations [1], [2], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43] have become indispensable in the fields of computer vision, graphics, and multimedia due to their ability to capture more comprehensive information beyond

simple spatial coordinates and normals. This includes important data such as confidence levels, radius, and timestamps.

In the data fusion process, surfels and points are projected onto incoming depth maps using a well-established sensor pose. To address the inherent uncertainties in depth estimation, Schöps et al. [4] suggest utilizing a depth interval that accommodates minor discrepancies in projection measurements. During this projection, correspondences are classified into three categories—supported, conflicting, or occluded—based on their spatial relations to the defined depth interval. Verified correspondences are then fused into a new surfel or point representation, typically through a weighted averaging function such as Gaussian weighting. Over time, the fused model is refined by removing conflicting surfels or points, driven by confidence levels and temporal constraints [2], [4].

Exploring a different approach, non-projective methods such as Gaussian Fusion [42] and Cyclical Fusion [1] have been developed, grounded in optimal transport theory to enhance the detailed reconstruction of indoor scenes. While both methods utilize the L^2-Wasserstein distance metric, they differ significantly in their optimization strategies. Gaussian Fusion [42] uses displacement interpolation based on geodesics in the L^2-Wasserstein space to establish point-to-point correspondences derived from the optimal transport plan. In contrast, Cyclical Fusion [1] constructs point-to-point correspondences using the principle of Cyclical Monotonicity, achieving quadratic time complexity compared to the cubic complexity associated with Gaussian Fusion [42].

Recent advancements have also led to the development of Gaussian splatting methods in both 3D [44] and 2D [45], significantly enhancing the rendering quality of 3D models. Simultaneously, transformer-based approaches have shown impressive outcomes in high-quality point cloud reconstruction [17] and multi-space point geometry compression [18]. However, it is essential to recognize that these state-of-the-art methods [11], [12], [17], [18], [46] are often computationally intensive and require substantial memory resources. Specifically, they necessitate a high thermal design power (TDP) modern GPU with significant memory capacity to effectively facilitate the reconstruction of a single scene.

Watertight Surface Reconstruction using Geometry-based Techniques encompass Poisson surface reconstruction [47], [48], [49], [50] and Delaunay-based approaches [51], [52], [53], [54], [55]. These methods provide complementary solutions to the challenges associated with point cloud reconstruction and fusion. The main objective of surface reconstruction is to create a watertight digital representation of physical shapes derived from scanned or reconstructed point clouds, which often contain defects [51], [56]. This aim distinguishes our work as it focuses on the fusion and refinement of point clouds rather than solely creating surfaces.

Poisson surface reconstruction addresses the reconstruction problem by solving a Poisson equation, which results in watertight surfaces. This method has been effectively adapted for large-scale point clouds across various applications, such as 3D object reconstruction and shape analysis [47], [48]. However, it is recognized for its computational intensity and high resource demands. To overcome these limitations, a distributed version of Poisson surface reconstruction has been introduced [49], enabling efficient operation on clustered computational resources.

Likewise, Delaunay-based methods successfully achieve watertight surface reconstruction, but they encounter similar computational challenges. A significant advancement in this area is the BallMerge technique [55], which performs up to two rapid linear-time passes over the Delaunay complex to generate the surface. This technique is an order of magnitude faster than current state-of-the-art methods, while also maintaining competitive memory usage, often achieving superior surface quality. These methodologies have the potential to complement our work, demonstrating that effective integration could further enhance the quality of reconstructed surfaces.

In this context, we propose a Manifold Embedding framework, which stands out as a non-projective technique compared to those previously discussed in the literature [2], [6], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [43]. This framework is designed to address the challenges posed by conventional methods, offering a novel approach to the fusion and reconstruction of point clouds.

III. Preliminaries

This section outlines the symbols, notations, and definitions used throughout this paper, providing a clear introduction to our mathematical symbols. For example, the notation $\langle , \rangle$ represents the inner product, while $|\cdot|$ denotes the Euclidean norm for vectors. In the context of sets, the notations $|K|$ and $|V|$ represent the number of elements in sets K and V, respectively. Additionally, the notation $\mathbb{R}^3$ indicates three-dimensional Euclidean space.

At time t, the input point cloud, generated from a depth map (RGB-D stream), is denoted as $K^t = \{p_i \in \mathbb{R}^3\}$. Here, each p_i represents the 3D coordinates of the sampling points. Each sampling point p_i is associated with a unit normal vector $n_i \in \mathbb{R}^3$, calculated from the depth map using finite differential techniques. It is also linked to a neighborhood radius $r_i \in \mathbb{R}^+$, which reflects the average distance to the 20 nearest points.

The accumulated model at time t is represented as

$$G^t \approx (G^{t-1} \cup K^t), \tag{1}$$

with the initialization $G^0 = K^0$. This means that the accumulated point cloud is the union of G^{t-1} and K^t, while excluding parts of the current frame K^t that correspond to the previously accumulated model G^{t-1}. This correspondence is verified and updated throughout the process. The main goal of this work is to efficiently establish the dense correspondence between the input point cloud K^t and the accumulated model G^{t-1}, while validating and updating the respective segment of the accumulated model.

To define the signed distance function from an unknown coordinate $x \in \mathbb{R}^3$ to the tangent plane at point p_i, we use the expression

$$\langle x - p_i, n_i \rangle. \tag{2}$$

By computing the weighted average of all signed distance functions for the input point cloud K^t at time t, we derive the implicit function $F(x)$, as further detailed in (9) to (20). The null space of $F(x)$, denoted as $Null(F)$, serves as an effective approximation of the input surface $\mathcal{S}$ at time t.

We construct the local surface using voxels, which are defined as local geometric units $V_k \in K^t$. These voxels contain 3D points p_i arranged with adaptive resolution. The voxel resolution is proportional to the average neighborhood radius, expressed as

$$\frac{1}{|K^t|}\sum_{i=1}^{|K^t|} r_i. \tag{3}$$

The center of each voxel is denoted by

$$\overline{p_k} = \frac{1}{|V_k|}\sum_{i=1}^{|V_k|} p_i, \tag{4}$$

while the center normal vector is given by

$$\overline{n_k} = \frac{1}{|V_k|}\sum_{i=1}^{|V_k|} n_i. \tag{5}$$

The voxel radius is represented as

$$r_k = \max(|p_i - \overline{p_k}|). \tag{6}$$

The nearest point from the accumulated point cloud G^{t-1} corresponding to the kth voxel V_k is represented as p_j^k. The update increment along the normal vector n_j^k of the point p_j^k is denoted as

$$\Delta p_j^k = d_j^k \cdot n_j^k, \tag{7}$$

where

$$d_j^k = \langle \overline{p_k} - p_j^k, n_j^k \rangle \tag{8}$$

represents the signed projected distance of the vector $\overrightarrow{p_j^k \overline{p_k}}$ onto n_j^k.

Verification and updating are crucial components of the fusion process. The verification filter is designed to validate adjacent sampling points $p_j^k \in G^{t-1}$ and voxels $V_k \in K^t$. It ensures that the inner product $\langle n_j^k, \overline{n_k} \rangle \approx 1$ holds with a certain probability, as shown in Fig. 2(b). This criterion is essential for confirming that the corresponding point is not an outlier. Additionally, the updating probability filter helps retain the increment of the update with a specified probability. This ensures that the modified point closely approximates the surface, as depicted in Fig. 2(c). The black curve in these illustrations represents the null space $Null(F)$ of the implicit function F, which is explored further in Figs. 4 and 2(d).

The central thesis of this paper centers on the concept of manifold representation, which involves representing a smooth surface using discrete sets of points. We consider a collection of sample points $\{p_i \in \mathbb{R}^3\}$ collected through a sensor, each associated with corresponding normal vectors $\{n_i \in \mathbb{R}^3\}$. These sample points are derived from an unknown surface $\mathcal{S}$. The primary objective is to develop a method for directly representing and reconstructing the surface from these sample points. We postulate that the surface can be characterized by the function $f(x)$ over an arbitrary parameter domain, with observed values f_i being approximated using the moving least-squares (MLS) method. Consequently, the surface function $f(x)$ is fitted using an approximated function $\hat{f}(x)$ within the three-dimensional polynomial space Π_l^3 of total degree l.

1) Formulate the Weighted Least-Squares Problem: The process begins by formulating the weighted least-squares problem. The goal is to find a function $\hat{f}(x)$ that minimizes the weighted sum of squared differences between the observed values f_i and the values predicted by $\hat{f}(x)$ at the points p_i:

$$f(x) = \arg\min_{\hat{f}\in\Pi_l^3} \sum_{i=1}^{N} \theta(|x - p_i|)|\hat{f}(p_i) - f_i|^2, \tag{9}$$

where $\theta(|x - p_i|)$ is a weight function that depends on the distance between x and p_i.

2) Define the Polynomial Approximation: Next, the approximation function $\hat{f}(x)$ is defined in terms of a polynomial basis. Let $\boldsymbol{g}(x)$ be the polynomial basis vector and $\boldsymbol{c}(x)$ be the unknown coefficient vector. The approximation $\hat{f}(x)$ is expressed as:

$$\hat{f}(x) = \boldsymbol{g}^T(x)\boldsymbol{c}(x) = \sum_{j=1}^{M} g_j(x)c_j(x), \tag{10}$$

where $M = (3 + l)!/3!l!$ is the number of coefficients, with l being the polynomial degree.

3) Reformulate the Objective Function: By substituting $\hat{f}(x)$ from (10) into (9), the objective function becomes:

$$f(x) = \arg\min_{\boldsymbol{c}} \sum_{i=1}^{N} \theta(|x - p_i|)|\boldsymbol{g}^T(p_i)\boldsymbol{c}(x) - f_i|^2. \tag{11}$$

4) Matrix Formulation: To solve for $\boldsymbol{c}(x)$, the problem is converted into a matrix form. Let Θ be the diagonal matrix with $\theta^{\frac{1}{2}}(|x - p_i|)$ as diagonal elements, and $\boldsymbol{G}$ be the matrix with rows $\boldsymbol{g}^T(p_i)$. The resulting linear system is:

$$\Theta\boldsymbol{G}\boldsymbol{c}(x) = \Theta\boldsymbol{f}, \tag{12}$$

where $\boldsymbol{f}$ is the vector of observed values $[f_1, f_2, \ldots, f_N]^T$.

5) Solution for Coefficients: The coefficients $\boldsymbol{c}(x)$ are obtained by solving the normal equations:

$$\Theta\boldsymbol{G}^T\boldsymbol{G}\boldsymbol{c}(x) = \Theta\boldsymbol{G}^T\boldsymbol{f}. \tag{13}$$

If $\Theta\boldsymbol{G}^T\boldsymbol{G}$ is not singular, then

$$\boldsymbol{c}(x) = (\Theta\boldsymbol{G}^T\boldsymbol{G})^{-1}\Theta\boldsymbol{G}^T\boldsymbol{f}. \tag{14}$$

6) Implicit Moving Least-Squares Surface: The implicit moving least-squares surface is defined by the null space of a signed distance field based on the moving least-squares method. To achieve both interpolation and approximation, Shen et al. [57] proposed the following weight function: $\theta(r) = 1/\left(r^2 + \epsilon^2\right)$.

Notice that $\epsilon = 0$ leads to singularity at $r = 0$.

7) Incorporate Normal Vector Constraints: Shen et al. [57] reconstructed the surface using both position constraints f_i and normal vector constraints $\langle n_i, x - p_i \rangle$:

$$\Theta\boldsymbol{G}\boldsymbol{c}(x) = \Theta\widetilde{\boldsymbol{f}}, \tag{15}$$

where $\widetilde{f}_i(x) = f_i + \langle n_i, x - p_i \rangle$.

8) Simplification for Linear Basis: For simplicity, let $g(x) = [1]$, so $f(x) = c(x)$ becomes a level set value. The fitting equation then simplifies to:

$$\left(\sum_{i=1}^{N} \theta^{\frac{1}{2}}(|x - p_i|) \right) f(x) = \sum_{i=1}^{N} \theta^{\frac{1}{2}}(|x - p_i|) \widetilde{f}_i(x). \quad (16)$$

9) Interpretation and Extension to Polygons: The fitting equation can be interpreted as predicting the weighted average value of each $\widetilde{f}_i(x)$. When reconstructing meshes from a polygon soup, point constraints are replaced with polygon constraints. Thus, the bracket term in (16) and the $\widetilde{f}_i(x)$ terms on the right are replaced by integrals over the polygon [57], [58]:

$$f(x) = \frac{\sum_i \int_{Poly_i} \theta(|x - p|)(f_i + \langle n_i, x - p_i \rangle) dp}{\sum_i \int_{Poly_i} \theta(|x - p|) dp}, \quad (17)$$

where $Poly_i$ denotes the ith polygon, and p is the integral variable on the polygon.

10) Uniform Implicit Moving Least-Squares Surface: To further simplify (17), let $f_i = 0$, then $\widetilde{f}_i(x) = \langle n_i, x - p_i \rangle$. In this simplest case, Kolluri [59] defined the uniform implicit moving least-squares surface function, proving it to be an ideal approximation of the signed distance function of the sampling surface. The implicit function $F(x)$ is defined as:

$$F(x) = \frac{\sum_{p_i \in \mathcal{S}} \theta(|x - p_i|) \langle x - p_i, n_i \rangle}{\sum_{p_i \in \mathcal{S}} \theta(|x - p_i|)}, \quad (18)$$

where $\theta(|x - p_i|) = \exp(-|x - p_i|^2 / r^2)/A_i$ is the average Gaussian weight, and A_i is the number of sampled points within a sphere centered at p_i with radius r.

The implicit moving least-squares (MLS) surface function is derived from (9) to (18), enabling a transition from polygonal to point surface representation. Building on this foundation, Liu et al. [15] introduced IMLSNets, an advanced reconstruction method that utilizes a deep implicit moving least-squares function. This approach effectively models smooth surfaces using point cloud data [60] and incorporates point-based moving least-squares (MLS) interpolation to define local implicit functions within the narrowband region of the point cloud during the construction process.

A key aspect of Liu et al.'s work is the application of the implicit moving least-squares surface function to map spatial points within the narrow region to a null space surface. This process generates a signed distance value characterized by a weighted mixture of oriented planes derived from the signed distances to nearby points. As a result, the null space surface is extracted, leading to a smooth and continuous representation of a 3D shape.

In contrast to IMLSNets [15], our framework does not rely on neural networks for point prediction. Instead, it employs the implicit moving least-squares surface function to guide the integration of points and surfaces. This non-trained fusion approach enhances the adaptability of our method, making it particularly suitable for prototype scanning systems aimed at reconstructing real-world environments. While IMLSNets have demonstrated impressive reconstruction performance on virtual datasets, they encounter challenges when applied to real-world scenarios.

IV. Method

This section provides a clear and detailed description of the proposed framework, focusing on the geometric representation of the surface. We begin with a brief overview of the voxel generation process [61], [62]. In contrast to traditional methods, our approach utilizes a point-to-surface fusion technique instead of a point-to-point framework.

Afterward, we use an implicit moving least-squares (MLS) surface function as a guiding mechanism. Alongside this, we implement a verification-updating probability filter to facilitate the fusion of adjacent points once the voxels and their corresponding neighboring points have been identified.

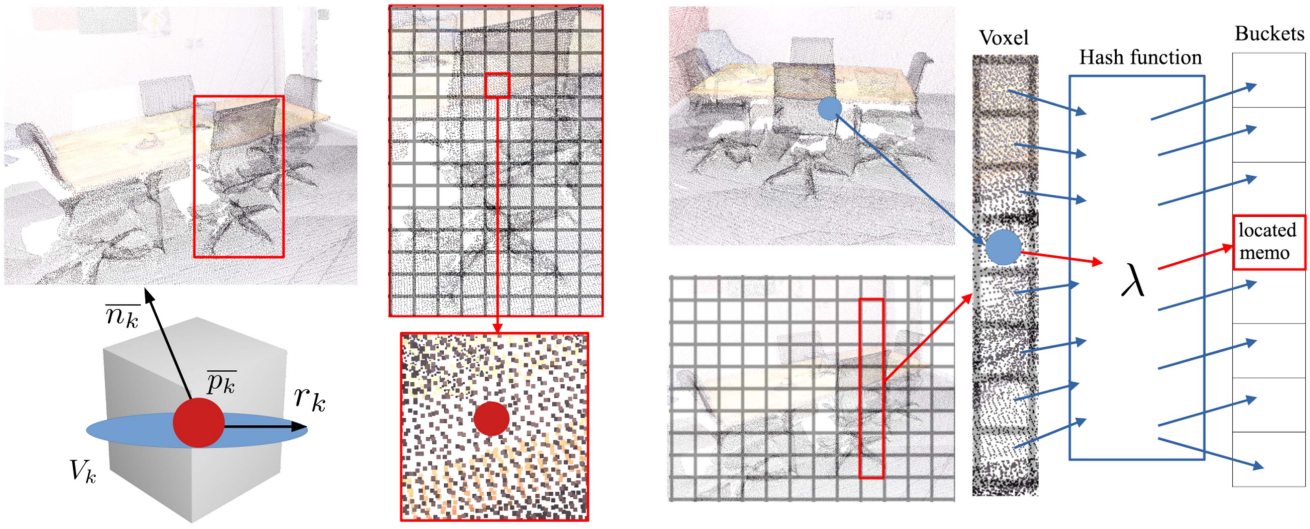
A. Manifold Voxel

To facilitate rapid neighboring searches between points and voxels, we implement hash indexing for spatial coordinates. First, the point cloud K^t is voxelized from the input frame [23], [61]. After this step, we perform a hash indexing search to find neighboring points within the accumulated model G^{t-1} that correspond to each voxel (see Fig. 3).

Our fusion methodology is distinct from previous approaches. Once we establish the relationship between a point and its corresponding voxel, we utilize the implicit moving least-squares surface function along with the Verification-Updating Probability Filter to guide the fusion process. This strategy bypasses the challenges of determining point-to-point correspondences based on geometric constraints. As a result, it effectively addresses the technical difficulties associated with achieving dense correspondences and ultimately enables faster 3D fusion.

The sampling points p_i, which reside within the neighborhood, define the local surface. We utilize these sampling points within the voxel to effectively characterize this local surface. The moving least-squares (MLS) surface offers several advantages and is particularly effective in managing perturbations related to depth and pose uncertainties. Therefore, we adopt the MLS surface for the geometric representation of the local surface. However, it is important to note that projecting the MLS surface requires frequent calculations of the projection reference surface, resulting in a significant increase in computational time as the number of sampling points rises. Consequently, projecting the MLS surface becomes impractical for dense samplings. To overcome this limitation, we use the implicit MLS surface function $F(x)$ proposed by Kolluri [59] to represent the local surface more efficiently.

The design objective of the Manifold Voxel is to effectively encapsulate local geometry and facilitate efficient neighboring searches. For local geometry representation, we rely on the MLS surface [59] to characterize the local surface, which is smooth and can be represented through a collection of sample points contained within the voxel. In this study, we employ a modified version of $F(x)$ (18), centered on the voxel, to represent the MLS surface. Alongside the voxel center $\overline{p_k}$, we define the



(a) Point cloud is voxelized

(b) Blue point searches nearby voxel in hash style

Fig. 3. *Manifold Voxel.* (a) Let $\overline{p_k}$ represents the center of the voxel, $\overline{n_k}$ denotes the center normal vectors, and $r_k = \max(|p_i - \overline{p_k}|)$ indicates the voxel radius; (b) A hash function is employed to store voxels in memory buckets, allowing the blue point to quickly access these buckets (the 2D scaled grids are provided for visualization of the voxels).

center normal vectors $\overline{n_k}$ and the voxel radius r_k, as illustrated in Fig. 3(a).

To enhance the efficiency of neighboring searches, we implement a hash function that organizes voxels into memory buckets. This approach is necessary because KD-tree-based nearest neighbor searches can become bottlenecks in dense point cloud scenarios. To address the challenges associated with costly nearest-neighbor searches, our voxel representation adopts a voxel hashing strategy. We store sample points in a voxel that conforms to a predefined resolution using a hash function (as shown in Fig. 3(b)). A rapid point-to-voxel search can be performed between a point and the neighboring voxel (illustrated in Fig. 3(b)). Typically, this search progresses through nearby voxels by advancing two voxel size steps in 3D space. Regarding voxel resolution, we configure it to be proportional to the average radius of the point cloud, ensuring adaptability to a wide range of scene variations.

The next subsection will elaborate on the proposed fusion method, guided by the implicit moving least-squares surface function.

B. Manifold Embedding Via IMLS

The implicit function $F(x)$ is calculated as the weighted average of all the signed distance functions derived from the input point cloud K^t at time t. The null space $Null(F)$ of $F(x)$ serves as an effective approximation of the underlying surface $\mathcal{S}$ at that time t.

$$F(x) = \frac{\sum_{p_i \in K^t} \theta(|x - p_i|, r_i)\langle x - p_i, n_i\rangle}{\sum_{p_i \in K^t} \theta(|x - p_i|, r_i)}, \tag{19}$$

where $\theta(|x - p_i|, r_i) = \exp(-|x - p_i|^2 / r_i^2)$ is the weight function. Kolluri has shown that when sampling is uniform in the point cloud K^t, the implicit moving least-squares surface function F accurately represents the original sampled surface $\mathcal{S}$ in terms of both geometry and topology. Additionally, F serves as a reliable approximation of the signed distance function of the original surface [15], [59].

Considering the weight function rapidly decays as x moves away from p_i, we focus exclusively on the sampling points within the same voxel instead of processing the entire point cloud K^t. This focused approach significantly enhances the computational efficiency of $F(x)$. Therefore, (19) can be reformulated as follows:

$$F(x) = \frac{\sum_{p_i \in V_k} \theta(|x - p_i|, r_k)\langle x - p_i, n_i\rangle}{\sum_{p_i \in V_k} \theta(|x - p_i|, r_k)}, \tag{20}$$

where $V_k = \{p_i \mid i = 1, \ldots, n\}$ represents a voxel within the point cloud K^t, which indicates that the voxel V_k contains n sampling points. Let $\overline{p_k}$ denote the center of the voxel, and define the maximum distance from any point in V_k to this center as the voxel radius r_k.

The null space $Null(F)$, derived from the function $F(x)$ as shown in (20), exists within a narrow band region. Minimizing $F(x)$ aids in the smooth reconstruction of point clouds. The implicit moving least-squares function $F(x)$ is particularly effective at addressing uncertainty issues. Our objective, as outlined by (20), is to achieve a smooth fusion of point clouds. Specifically, we aim to merge the sampling points of G^{t-1} smoothly, utilizing the implicit surface function $F(x)$ based on the input sampling point cloud K^t.

In the accumulated model G^{t-1}, each sampling point $p_i \in \mathbb{R}^3$ is associated with a unit normal vector $n_i \in \mathbb{R}^3$. When G^{t-1} aligns with the pose of the input sampling point cloud K^t at time t, we can identify the nearby point set $\{p_1^k, p_2^k, p_3^k, \ldots, p_m^k\} \in G^{t-1}$, that corresponds to the input voxel $V_k \in K^t$ through a hashing index (as illustrated in Fig. 3(b)). In this notation, the superscript k of p_j^k indicates that p_j^k is a nearby point related

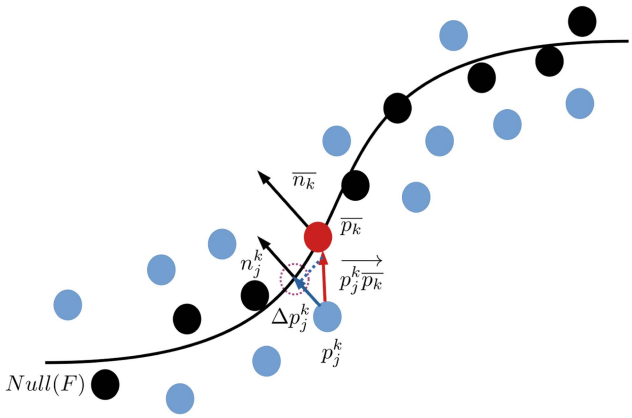


Fig. 4. *Implicit Moving Least-Squares Guided Fusion.* The blue points p_j^k are derived from the accumulated point cloud G^{t-1} and correspond to the kth voxel V_k in the input frame K^t (as illustrated in Fig. 3, which demonstrates how to identify the voxel using a hash-style approach). The center of each voxel is denoted by $\overline{p_k}$. The increment $\Delta p_j^k = d_j^k \cdot n_j^k$ is applied along the normal vector n_j^k associated with p_j^k. The black curve represents the null space $Null(F)$ of the implicit function $F(x)$, which approximates the smooth manifold $\mathcal{S}$.

to the k-th voxel V_k, which is distinct from the points $p_i \in V_k$. Since $F(x)$ approximates the signed distance function of the original surface, the optimization process focuses on embedding the point p_j^k into a smooth manifold.

$$\arg \min_{\alpha \in [0,1]} |F(p_j^k + \alpha \cdot \Delta p_j^k)|. \tag{21}$$

The update optimization process for fusion is outlined in (21). In this equation, Δp_j^k represents the update increment along the normal vector n_j^k of the point p_j^k. The term d_j^k indicates the signed projected distance from the vector $\overrightarrow{p_j^k\overline{p_k}}$ onto the normal vector n_j^k. The expression $p_j^k + \alpha \cdot \Delta p_j^k$ defines the direction of the update for p_j^k, where α takes values from 0 to 1. Equation (21) shows that the implicit moving least-squares function F guides the update of p_j^k toward the desired direction. As F approaches zero, the position of the embedded point gets closer to the sampling surface, as depicted in Fig. 4. The optimization of this specified direction in (21) can be efficiently accomplished through iterative processes, as detailed in Algorithm 1.

The implicit moving least-squares (IMLS) guided fusion methodology is composed of a systematic sequence of steps. Initially, the IMLS function F provides an accurate approximation of the original surface's signed distance function, as demonstrated in previous studies [15], [59]. This function offers a smooth representation of the local surface corresponding to the input sampling K^t through its null space, noted as $Null(F)$. Guided by F, the adjacent point set within the accumulated model G^{t-1} undergoes a smoothing process using our Manifold Embedding framework. This smooth fusion is achieved through iterative optimization aimed at minimizing (21), as illustrated in Fig. 4. In this figure, the black dots represent the sampling points located within voxel V_k, while the red point $\overline{p_k}$ indicates the center of the voxel. The average normal vector is represented by $\overline{n_k}$. The black curve shows the null space $Null(F)$ of the implicit function F, which serves as a smooth approximation of the local surface within the t-th input sampling point cloud K^t. Additionally, the blue dots represent the nearby sampling points $\{p_1^k, p_2^k, p_3^k, \ldots, p_m^k\} \in G^{t-1}$ around V_k. Furthermore, the black arrow n_j^k signifies the unit normal vector associated with the point p_j^k. The signed projected distance of the vector $\overrightarrow{p_j^k\overline{p_k}}$ onto n_j^k is denoted by d_j^k. Consequently, the update increment is defined as $\Delta p_j^k = d_j^k \cdot n_j^k$, which reflects the adjustment along the normal vector n_j^k for the point p_j^k. In the surface fusion process, the point set is updated in a specified direction, expressed as $p_j^k + \alpha \cdot \Delta p_j^k$, where $\alpha \in [0, 1]$. When $\alpha = 1$, the update moves directly towards the voxel's center $\overline{p_k}$ along the normal vector n_j^k (indicated by the purple dashed line). Conversely, when $\alpha = 0$, the point p_j^k remains in its original position. Thus, under the guidance of F, we achieve a smooth fusion that results in a coherent manifold through a series of steps. This surface fusion technique operates with high computational efficiency, allowing us to avoid the need for dense matching.

The IMLS method effectively addresses challenges related to uncertainty interference, improving the quality of the point cloud by smoothing out noise points and mitigating the effects of multi-layer surfaces caused by pose uncertainty. A critical aspect of this method involves updating the points along their respective normal vectors, which highlights the importance of normal vectors in the fusion process.

Before executing the fusion, it is essential to verify that the sampling points in the accumulated model are oriented correctly according to the normal vectors. Specifically, the directions of the normal vectors for the adjacent sampling points in the accumulated model must be consistent with those of the local surface intended for fusion. Effective verification of the normal vectors can significantly prevent declines in reconstruction quality that result from inappropriate fusion. While it is expected that multiple samplings of the same surface will show similar normal vector directions, uncertainties can negatively affect some of these samples. Fortunately, these problematic samples can be efficiently excluded through proper normal vector verification.

However, establishing hard thresholds for normal vector verification can be challenging, particularly in environments with complex geometries. To address this issue, we propose a verification-updating two-step probability filter, which enhances both the accuracy and speed of 3D fusion. The next subsection will elaborate on how the verification-updating two-step probability filter facilitates rapid and precise 3D fusion.

C. Verification-Updating Two-Step Probability Filters

The fusion process can be effectively divided into two main stages: Verification and Updating. In the SurfelMeshing approach to fusion [4], projection points are categorized into three types: support points, occlusion points, and conflict points. The first stage, known as the Verification process, focuses on validating these projection points. Once the Verification is complete, the support points are updated using weighted values. This results in the development of a two-step filter that systematically validates and updates the sampling data (see Algorithm 1).

Verification probability filter: The Verification Probability Filter is designed to evaluate a set of adjacent sampling points $\{p_1^k, p_2^k, p_3^k, \ldots, p_m^k\}$ derived from a hash index. Its main objective is to determine whether these points serve as supporting points for the corresponding voxel.

The Verification process considers four possible scenarios:

1. The sampling point p_j^k and its corresponding adjacent voxel V_k from the input frame originate from the same scanning surface. In this case, p_j^k is a supporting point for voxel V_k, and V_k acts as a supporting voxel for point p_j^k.

2. The sampling point p_j^k is influenced by uncertainties related to pose and depth, classifying it as a perturbation point.

3. The adjacent voxel V_k on the input frame is also affected by uncertainties of pose and depth, making it a perturbation voxel.

4. Both the adjacent voxel V_k and the sampling point p_j^k from the K^t and G^{t-1} are compromised by these uncertainties, rendering them as perturbation points and voxels. This situation hinders their utility in the fusion process.

A single fusion operation cannot effectively differentiate between scenarios 2, 3, and 4. However, multiple iterations of fusion can identify perturbation voxels or points by evaluating the frequency of fusion occurrences and their associated confidence levels.

The primary aim of the Verification Probability Filter is to reduce computational load while focusing on confirming scenario 1. This is achieved by utilizing spatial position constraints (adjacent relationships) and normal vectors. Specifically, verification analyzes the cosine angle between the normal vectors.

For each point p_j^k with an associated normal vector n_j^k from the accumulated model, it is essential that the angle between this normal vector and the average normal vector $\overline{n_k}$ of the corresponding voxel V_k, derived from the input sampling frame K^t, is close to zero. Thus, the dot product $\langle n_j^k, \overline{n_k} \rangle$ should approach one.

We employed a Verification Probability Filter informed by the accept-reject sampling framework [63], which functions without predefined verification angle thresholds. In the following sections, we will elaborate on the specific components of the Verification Probability Filter.

1) Target Distribution Function: The target distribution for the Verification Probability Filter is designed to ensure that adjacent sampling points are preserved. Specifically, we aim for the condition $\langle n_j^k, \overline{n_k} \rangle \approx 1$ to hold true with a specified probability. To facilitate this goal, we define a new variable given by $\psi_j^k = \langle n_j^k, \overline{n_k} \rangle - 1$. With this in mind, we formulate the target distribution as follows:

$$f_1(\psi_j^k) = \frac{1}{\sqrt{2\pi}\sigma_1} exp\left(-\frac{(\psi_j^k - 0)^2}{2\sigma_1^2}\right), \tag{22}$$

The target distribution is modeled as a normal distribution with a mean of zero and a standard deviation set to σ_1. For our experiments, we configure σ_1 to 0.5.

2) Uniform Sampling Verification: Consider a point p_j^k located within the accumulation model G^{t-1} and its corresponding adjacent voxel V_k, which is sourced from the input sampling frame K^t. The verification process for the sampling point is as follows: If the normal vector n_j^k of the sampling point p_j^k meets the condition $\mu_1 \leq f_1(\psi_j^k)$, where μ_1 is drawn from a uniform distribution $\boldsymbol{U}(0, 1)$, then the sampling point will be retained for further processing. Conversely, if the normal vector n_j^k does not satisfy this condition, the sampling point will be excluded from the subsequent updating and optimization procedures.

Updating probability filters: The process of updating the probability filters begins when the normal vector n_j^k associated with the sampling point p_j^k meets the condition $\mu_1 \leq f_1(\psi_j^k)$. According to (21), this update procedure is guided by the implicit least-squares function F. The updating increment is defined as $\Delta p_j^k = d_j^k \cdot n_j^k$, where d_j^k represents the signed projection distance of the vector $\overrightarrow{p_j^k \overline{p_k}}$ onto n_j^k. The main goal of the updating probability filters is to ensure that the ratio of the projection distance to $|\overrightarrow{p_j^k \overline{p_k}}|$ approaches unity. The design of the updating probability filter is detailed as follows:

1) Target Distribution Function: As previously discussed, the optimization procedure outlined in (21) is focused on moving the sampling point along the normal vector n_j^k toward the voxel center $\overline{p_k}$. To retain the sampling points, the updating probability filter enforces a probabilistic condition such that $|\langle \overline{p_k} - p_j^k, n_j^k \rangle / |\overline{p_k} - p_j^k|| \approx 1$. To support this process, we define $\phi_j^k = |\langle \overline{p_k} - p_j^k, n_j^k \rangle / |\overline{p_k} - p_j^k|| - 1$. The target distribution function $f_2(\phi_j^k)$ follows a pattern similar to that in (22). This target distribution is modeled as a normal distribution with a mean of zero and a standard deviation of σ_2, which we have empirically set to $\sigma_2 = 0.5$.

2) Uniform Sampling Verification: For a neighboring sampling point p_j^k within the voxel V_k, we validate its inclusion through the probability filter by checking whether $\mu_2 \leq f_2(\phi_j^k)$, where μ_2 is a value drawn from a uniform distribution $\boldsymbol{U}(0, 1)$. If p_j^k fails to meet this condition, it will be excluded from the update optimization process. On the other hand, sampling points p_j^k that successfully pass through the updating probability filter will undergo iterative optimization according to (21).

For each input voxel V_k, if it has not yet been validated and updated, the points contained within the voxel will be integrated into the accumulation model as new verification-updating points p_i. Points that have previously undergone verification and are retained in the accumulation model will ultimately be output as the final fusion result.

In summary, the Verification-updating two-step probability filter presents two significant advantages. First, it maintains sampling points based on probabilistic criteria, thereby obviating the necessity for rigid threshold settings. This characteristic enhances the method's adaptability to varying surface manifolds. Second, unlike fixed thresholds, the Verification-updating two-step probability filter exhibits a notable degree of fault tolerance. Given that accurate estimation of normals amidst uncertainty represents a considerable challenge, the preservation of sampling points in a probabilistic framework effectively addresses the uncertainties associated with normal vector estimations, particularly when errors are present. This efficacy in managing estimation errors is further substantiated by the ablation

Algorithm 1: Manifold Embedding.

Input: *Manifold Voxel* V_k: $p_i, n_i, \overline{p_k}, \overline{n_k}$
Point and Normal: p_j^k, n_j^k
Output: *Embedded* $p_j^{'k}$ *or Original* p_j^k

Verification Probability Filter:
$\mu_1 \longleftarrow uniform\_distribution < float > (0,1)$;
$\psi_j^k \longleftarrow \langle n_j^k, \overline{n_k} \rangle - 1$;
$f_1(\psi_j^k) \longleftarrow \frac{1}{\sqrt{2\pi}\sigma_1} \exp(-\frac{{\psi_j^k}^2}{2\sigma_1^2})$;
if $\mu_1 \leq f_1(\psi_j^k)$ **then**
  Updating Probability Filter:
  $\mu_2 \longleftarrow uniform\_distribution < float > (0,1)$;
  $\phi_j^k \longleftarrow |\langle \overline{p_k} - p_j^k, n_j^k \rangle| / |\overline{p_k} - p_j^k| - 1$;
  $f_2(\phi_j^k) \longleftarrow \frac{1}{\sqrt{2\pi}\sigma_2} \exp(-\frac{{\phi_j^k}^2}{2\sigma_2^2})$;
  if $\mu_2 \leq f_2(\phi_j^k)$ **then**
    $F(x)_{min} \longleftarrow numeric\_limits < float > :: max()$
    $\alpha_{min} \longleftarrow 0$
    $\Delta p_j^k \longleftarrow \langle \overline{p_k} - p_j^k, n_j^k \rangle \cdot n_j^k$
    $r_k \longleftarrow max(|p_i - \overline{p_k}|)$
    for *step* in 0 to 10 **do**
      $\alpha \longleftarrow step * 0.1$
      $F(x)_{molecule} \longleftarrow 0$
      $F(x)_{denominator} \longleftarrow 0$
      $x \longleftarrow p_j^k + \alpha \cdot \Delta p_j^k$
      for p_i in V_k **do**
        $\theta \longleftarrow exp(-|x - p_i|^2 / r_k^2)$
        $F(x)_{molecule} \longleftarrow F(x)_{molecule} + \theta \cdot \langle x - p_i, n_i \rangle$
        $F(x)_{denominator} \longleftarrow F(x)_{denominator} + \theta$
      end for
      $F(x) \longleftarrow abs(F(x)_{molecule} / F(x)_{denominator})$
      if $F(x) < F(x)_{min}$ **then**
        $F(x)_{min} \longleftarrow F(x)$
        $\alpha_{min} \longleftarrow \alpha$
      end if
    end for
    if $\alpha_{min} > 0$ **then**
      $p_j^{'k} \longleftarrow p_j^k + \alpha_{min} \cdot \Delta p_j^k$
      return *embedded* $p_j^{'k}$
    else
      return *original* p_j^k
    end if
  end if
else
  return *original* p_j^k
end if

study provided in the Supplementary materials, specifically illustrated in Supply-Fig. 2 and detailed in Supply-Table I.

V. Experiment

The experiments were organized into three main components. The first component focused on implementing the **Manifold Embedding** framework, as proposed in this study, to function as the fusion backend for our prototype system. We utilized visual-inertial odometry to perform comprehensive 3D scanning in real-world environments, leveraging the ground-truth calibration data of the *Skull* object for ground-truth comparison experiments. In the second component, we demonstrated the versatility of the proposed fusion algorithm by conducting comparative experiments with various consumer-grade RGB-D cameras, specifically using the *FastCaMo-Real* dataset [6]. Finally, our methodology underwent a comparative analysis with contemporary learning-based approaches on the *Replica* dataset [64], with a focus on the temporal aspects of the mapping and fusion processes. The extensive results obtained from these experiments underscored the effectiveness of the proposed fusion algorithm. Access to the codes and datasets was graciously provided by the corresponding authors [1], [3], [4], [6], [11], [42]. The default parameters used for the other evaluated methodologies were taken from the original authors. Notably, the default parameters of **NICE-SLAM** [11] were found to yield state-of-the-art performance on the *Replica* dataset.

A. Metrics

The metrics used across different datasets exhibit slight variations. For the *Skull* in the real-world scan and the *Replica* dataset, we adopted metrics that closely align with those established in the original publications [11], [12]. Specifically, accuracy (measured in mm/cm) is defined as the average distance between sampled points and the nearest ground-truth point. Completeness, which is measured in cm for the *Replica* dataset, is characterized by the average distance between sampled points and the nearest point on the ground-truth. The Completeness Ratio quantifies the percentage of points within the ground-truth point that achieve completeness below designated thresholds; specifically, this threshold is set at 2 mm for the *Skull* and 5 cm for the *Replica* dataset. For the *FastCaMo-Real* dataset [6], we employed metrics as described in the original research, focusing on reconstruction quality. This encompasses both completeness and accuracy relative to the ground-truth surfaces. In evaluating accuracy, we calculated the RMS error exclusively over the overlapping (inlier) regions between the reconstructed and ground-truth surfaces, with a threshold for inliers set at 15 cm.

B. Prototype System Scanning Experiment

The sampling front end of this section is composed of two advanced sensors: the Intel RealSense™ Laser Camera L515 and the Intel RealSense™ Tracking Camera T265. The evaluation results obtained from the experimental setup are presented in Table I. This table illustrates that the proposed method significantly outperforms the optimal-transport-based fusion technique. Furthermore, the Manifold Embedding framework introduced in this research shows improved accuracy compared to the methodologies described in [42] and [1]. In fact, the techniques discussed here achieve much higher accuracy metrics than these existing approaches. The novel Manifold Embedding framework features a two-step filtering process, which consists of implicit moving least-squares guided fusion and verification-updating

TABLE I
QUALITATIVE COMPARISON ON *SKULL*.

	Com. Ratio%↑	Acc.↓
ElasticFusion [3]	63.5%	2.5mm
SurfelMeshing [4]	62.7%	2.0mm
GaussianFusion [42]	87.0%	1.6mm
CyclicalFusion [1]	72.0%	1.5mm
ManifoldEmbedding	**89.2%**	**1.3mm**

TABLE II
THE COMPARISON OF COMPLETENESS AND SURFACE ERROR ON *FASTCAMO-REAL* DATASET [6] (AVERAGE OVER 12 SCENES)

	Com.Ratio%↑	Acc.↓
ElasticFusion [3]	20.2%	7.9cm
BundleFusion [26]	23.3%	6.0cm
ROSEFusion [6]	77.8%	5.3cm
CyclicalFusion [1]	76.7%	3.7cm
ManifoldEmbedding	**79.3%**	**3.6cm**

TABLE III
RECONSTRUCTION RESULTS FOR THE *REPLICA* DATASET [64] (AVERAGE OVER 8 SCENES)

	Acc.↓	Com.↓	Com.Ratio%↑
iMAP [12]	6.95cm	5.33cm	66.60%
DI-Fusion [13]	19.40cm	10.19cm	72.96%
NICE-SLAM [11]	2.85cm	3.00cm	89.33%
ManifoldEmbedding	**2.11cm**	**2.80cm**	**89.41%**

TABLE IV
COMPUTATION & RUNTIME

	GPU	Mapping/Fusion↓
iMAP [12]	Morden GPU	448ms
NICE-SLAM [11]	RTX 3090	130ms
GaussianFusion [42]	RTX 4050	4000ms
CyclicalFusion [1]	RTX 4050	130ms
ManifoldEmbedding	RTX 4050	**50ms**

sampling. This innovative approach leads to a substantial enhancement in reconstruction accuracy, effectively demonstrating the efficacy of the proposed techniques.

C. FastCaMo-Real Experiment

To demonstrate the applicability of our proposed method across a range of consumer-grade RGB-D cameras, we conducted a series of comparative experiments utilizing the *FastCaMo-Real* dataset [6]. The Microsoft Azure Kinect™ DK is employed to capture indoor environments, allowing us to explore various scenarios that reflect real-world applications. The *FastCaMo-Real* dataset [6] was explicitly designed to encompass rapid camera motion and provides ground-truth data collected from an industrial-grade scanner. For our fusion comparison experiments, we used the ROSEFusion algorithm [6] to compute the input poses. We adhered to the evaluation protocols established in previous work [6], ensuring consistency and reliability in our assessments. Our method exhibited competitive reconstruction accuracy across all experiments, as shown in Table II, along with Supplementary Tables III and IV. The results are further illustrated in Fig. 5.

D. Replica Experiment

To compare our method with contemporary learning-based approaches, we conducted a comprehensive evaluation utilizing the *Replica* dataset [64]. This dataset provides a rich set of indoor scenarios that are ideal for testing various aspects of reconstruction and fusion. For our evaluation, we specifically employed the rendered RGB-D sequences supplied by the authors of NICE-SLAM [11] and iMAP [12], which offer high-quality input data for robust analysis. For pose estimation, we utilized visual odometry [65], specifically focusing on reprojection error as a quantifiable metric.

By leveraging the Manifold Embedding framework, our approach achieved state-of-the-art performance. The efficacy of our method is clearly illustrated in the results presented in Table III, Supplementary Table II, Fig. 6, Supplementary Fig. 3, and Fig. 4. These visualizations and numerical results highlight not only the accuracy of our method but also its ability to operate efficiently under various conditions.

Importantly, our approach excels in several key aspects: it demonstrates minimal power consumption, as it does not require training, and it features reduced runtime, making it suitable for real-time applications. Details regarding these efficiency gains are further elaborated in Table IV, which showcases our method's performance in practical scenarios. Overall, our evaluation confirms that our method stands out in terms of both accuracy and efficiency, providing a compelling solution for depth and pose uncertainties in 3D reconstruction.

VI. DISCUSSION

This section analyzes key aspects of the proposed method, highlighting its limitations and exploring potential directions for future research.

A. Limitations and Future Work

The proposed method presents several opportunities for enhancement that warrant exploration in future research endeavors:

Incorporating RGB Streams for Enhanced Rendering: Currently, the reconstruction process primarily relies on geometric data, while color representation has received relatively limited attention. The RGB stream plays a specific role in visual odometry and pose estimation. However, to achieve a more realistic and visually compelling representation of the scene, we propose extending our efforts to include 3D rendering and novel view synthesis—both critical components of comprehensive 3D reconstruction.

As of 2023 and 2024, Gaussian splatting and its variants [44], [45] have emerged as leading techniques in this field. Nevertheless, the successful implementation of these algorithms depends on the availability of accurate 3D points for effective initialization of the 3D Gaussians. In situations where 3D points are sparse and noisy—often a result of depth uncertainty—these algorithms may deliver suboptimal rendering outcomes. Fortunately, recent advancements have made significant progress in addressing this challenge. Notably, one study [66] introduces a densification method capable of generating high-quality point

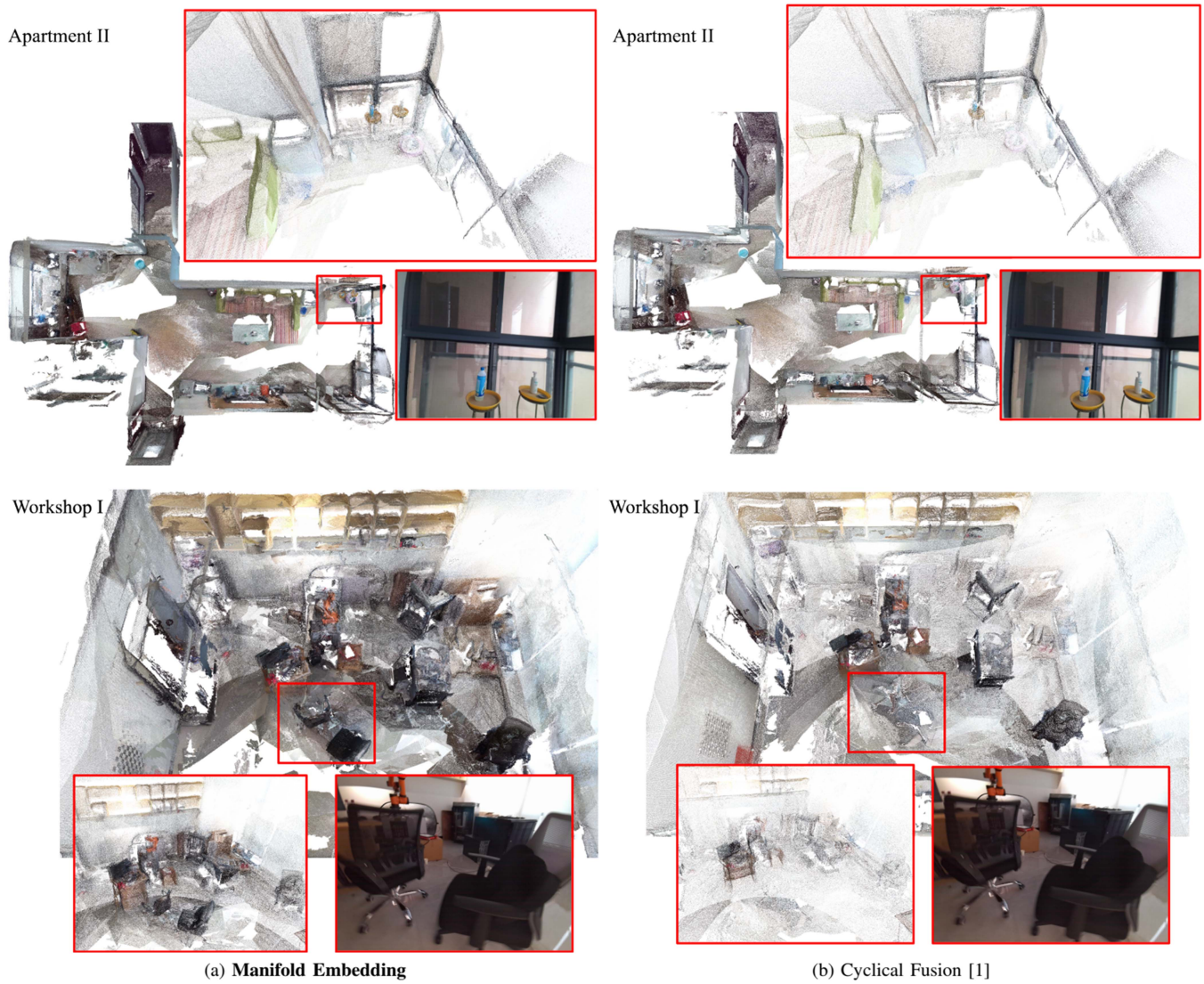


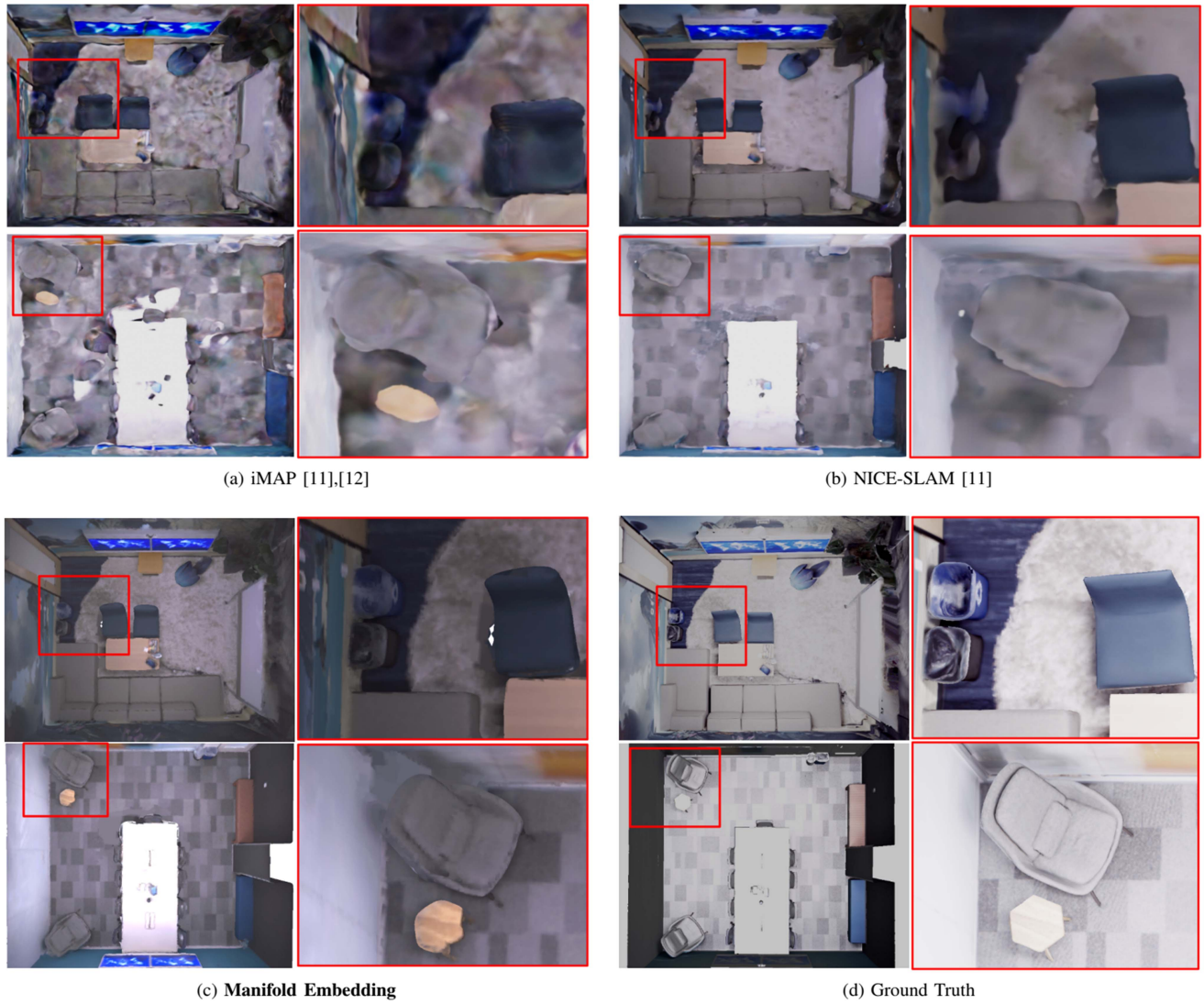
Fig. 5. *Point Cloud Reconstruction on FastCaMo-Real* [6] (Apartment II, Workshop I). The Manifold Embedding approach (left (a)) surpasses the Cyclical Fusion method [1] (right (b)) in capturing fine details, effectively reconstructing the thin table and black chair. However, the floor of Apartment II has some missing data due to limitations in the original scan, which may lead to a misleading smooth appearance in the mesh reconstruction. To clarify, we provide the raw reconstructed point cloud.

clouds from a single view, thus enhancing the reliability of 3D Gaussian initialization. Our approach has the potential to further improve point generation through multiview fusion, which could significantly enhance the overall quality of rendering.

Outdoor Scenes and Dynamic Scenes: The experiment is primarily designed for indoor environments, and we expect a significant decline in performance when applied to outdoor settings. This decline arises largely from the substantial reduction in quality that is characteristic of consumer RGB-D sensors. Ambient sunlight can interfere with the infrared light emitted by these sensors, such as Time-of-Flight-based sensors, Intel RealSense™, and Microsoft Azure Kinect™ DK, all of which were used in our experiments.

The vertical cavity surface emitting laser (VCSEL) and near-infrared (NIR) laser diodes utilized in these sensors have relatively low power compared to the intensity of sunlight. Consequently, the sensors often struggle to effectively distinguish between ambient light and laser reflections. This results in increased uncertainties in both depth and pose estimation (LiDAR-Inertial-Visual Odometry) when deployed in outdoor scenes, where measurement errors can vary from several centimeters to several meters. Such variations undermine the local smoothness assumption typically relied upon during indoor operations. We will further explore this issue in the section discussing failure cases and limitations. To address these challenges, we plan to enhance our methodology by integrating global geometries and leveraging outdoor LiDAR systems designed for autonomous driving, particularly to navigate the complexities of larger scenes.

Additionally, our current method is specifically tailored for static scenes and does not accommodate dynamic scenes directly. As a result, moving objects must be excluded from the



(a) iMAP [11],[12] (b) NICE-SLAM [11]

(c) **Manifold Embedding** (d) Ground Truth

Fig. 6. *Reconstruction Results on the Replica [64] Dataset.* Our method demonstrates notable improvements in indoor scene reconstruction using the *Replica* dataset [64], achieving highly detailed geometries with fewer artifacts, as seen with the small table beside the chair. Moreover, NICE-SLAM requires 24 GB of GPU memory for reconstruction.

scene prior to reconstruction within our existing framework. Tracking moving entities is fundamentally different from tracking static scenes, complicating pose estimation since simultaneous estimation is required for dynamic elements. Moreover, the occlusion of moving objects presents another significant challenge that our current approach cannot address. The interaction between deformable and moving objects represents one of the more complex scenarios we encounter. One potential solution is to reconstruct the static scene independently while handling the dynamic and deformable elements separately. Future research could focus on developing advanced methodologies that effectively manage dynamic scenes and moving objects in conjunction with our existing framework.

Failure Cases and Limitations: Our experimental results demonstrate that our methods can effectively manage the majority of indoor scenes, particularly those with intricate details related to objects on the centimeter scale. This success is due to the relatively small uncertainties inherent in indoor environments, which are typically confined to centimeters. We have calibrated the voxel resolution to align with the average point radius, also approximately on the centimeter scale, and set the voxel search radius to 2. Consequently, the reconstruction process leverages local geometries within a few centimeters, accommodating uncertainties such as depth and pose errors that remain similarly constrained to the centimeter scale.

In contrast, uncertainties encountered in outdoor scenes are generally much larger, often on the order of meters. As a result, the hash voxel search mechanism used for identifying nearby voxels may not perform effectively in these outdoor contexts. While the local smooth surface assumption is valid when applied to voxels at the centimeter scale, it becomes untenable when extended to voxels at the meter scale. Therefore, merely increasing the voxel size to meters does not adequately address the significant uncertainties present in outdoor environments.

The identified failure cases primarily stem from uncertainties at the meter scale and the vast dimensions of outdoor scenes, which may violate the assumption of local smooth surfaces. Furthermore, the application of Manifold Embedding, which focuses exclusively on local geometries, has proven inadequate for large-scale scenes characterized by meter-scale uncertainties. This emphasizes the need to incorporate global geometrical considerations, complicating our approach further.

Future research will focus on developing innovative methodologies designed to effectively manage large-scale scenes while addressing the inherent uncertainties found in outdoor environments. This research will prioritize the integration of both local and global geometrical considerations, ultimately enhancing our capacity to tackle the complexities associated with outdoor scene reconstruction.

B. Conclusion

This paper addresses the challenges involved in estimating point-to-point or pixel dense correspondences by introducing a novel point-to-surface fusion method based on Manifold Embedding. We present two innovative technologies: (1) **Implicit Moving Least-Squares Guided Fusion** and (2) **Verification-Updating Two-Step Probability Filter**.

Through extensive comparative experiments, we demonstrate the effectiveness of our proposed approach. The paper provides a detailed exploration of how to embed points into smooth manifolds and achieve point-to-surface fusion using implicit moving least-squares. Both experimental and theoretical results indicate that our method effectively overcomes the technical limitations associated with point-to-point or pixel dense correspondence estimation.

Our proposed method enables the rapid and seamless reconstruction of complex indoor scenes. Consequently, this research contributes a robust tool for 3D reconstruction and fusion applications across various domains.

ACKNOWLEDGMENT

The authors extend their gratitude to Prof. Kai Xu, the authors of the ROSEFusion, who provided the *FastCaMo-Real* dataset.

REFERENCES

[1] D. Chen, Z. Tang, and Y. Liu, "Cyclical fusion: Accurate 3D reconstruction via cyclical monotonicity," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 3955–3964.

[2] M. Keller et al., "Real-time 3D reconstruction in dynamic scenes using point-based fusion," in *Proc. 2013 Int. Conf. 3D Vis.*, 2013, pp. 1–8.

[3] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1697–1716, 2016.

[4] T. Schöps, T. Sattler, and M. Pollefeys, "Surfelmeshing: Online surfel-based mesh reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2494–2507, Oct. 2020.

[5] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. 2012 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[6] J. Zhang, C. Zhu, L. Zheng, and K. Xu, "Rosefusion: Random optimization for online dense reconstruction under fast camera motion," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–17, 2021.

[7] C. Villani, *Optimal Transport: Old and New*, vol. 338. Berlin, Germany: Springer Science & Business Media, 2008.

[8] Z. Feng, L. Yang, P. Guo, and B. Li, "Cvrecon: Rethinking 3D geometric feature learning for neural reconstruction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 17750–17760.

[9] D. Muhle, L. Koestler, K. M. Jatavallabhula, and D. Cremers, "Learning correspondence uncertainty via differentiable nonlinear least squares," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 13102–13112.

[10] B. Roessle and M. Nießner, "End2end multi-view feature matching with differentiable pose optimization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 477–487.

[11] Z. Zhu et al., "Nice-SLAM: Neural implicit scalable encoding for SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12786–12796.

[12] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "iMap: Implicit mapping and positioning in real-time," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6229–6238.

[13] J. Huang, S.-S. Huang, H. Song, and S.-M. Hu, "Di-fusion: Online implicit 3D reconstruction with deep priors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8932–8941.

[14] S. Weder, J. L. Schonberger, M. Pollefeys, and M. R. Oswald, "Neuralfusion: Online depth fusion in latent space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3162–3172.

[15] S.-L. Liu et al., "Deep implicit moving least-squares functions for 3D reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1788–1797.

[16] S. Weder, J. Schonberger, M. Pollefeys, and M. R. Oswald, "Routedfusion: Learning real-time depth map fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4887–4897.

[17] H. Tian, Z. Qin, R. Yi, C. Zhu, and K. Xu, "Tensorformer: Normalized matrix attention transformer for high-quality point cloud reconstruction," *IEEE Trans. Multimedia*, vol. 27, pp. 718–730, 2025.

[18] W. Shen, B. Zhang, H. Xu, X. Li, and J. Wu, "Multi-space point geometry compression with progressive relation-aware transformer," *IEEE Trans. Multimedia*, vol. 26, pp. 8969–8980, 2024.

[19] Z. Wang et al., "Learning to detect head movement in unconstrained remote gaze estimation in the wild," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 3443–3452.

[20] P. Chen et al., "Point-to-box network for accurate object detection via single point supervision," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 51–67.

[21] J. Gu et al., "Msinet: Twins contrastive search of multi-scale interaction for object reid," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 19243–19253.

[22] R. A. Newcombe et al., "Kinectfusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, 2011, pp. 127–136.

[23] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3D reconstruction at scale using voxel hashing," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–11, 2013.

[24] Q.-Y. Zhou and V. Koltun, "Dense scene reconstruction with points of interest," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–8, 2013.

[25] Q.-Y. Zhou, S. Miller, and V. Koltun, "Elastic fragments for dense scene reconstruction," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 473–480.

[26] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration," *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 1.

[27] F. Steinbrücker, J. Sturm, and D. Cremers, "Volumetric 3D mapping in real-time on a CPU," in *Proc. 2014 IEEE Int. Conf. Robot. Automat.*, 2014, pp. 2021–2028.

[28] O. Kähler, V. Prisacariu, J. Valentin, and D. Murray, "Hierarchical voxel block hashing for efficient integration of depth images," *IEEE Robot. Automat. Lett.*, vol. 1, no. 1, pp. 192–197, Jan. 2016.

[29] M. Klingensmith, I. Dryanovski, S. S. Srinivasa, and J. Xiao, "Chisel: Real time large scale 3D reconstruction onboard a mobile device using spatially hashed signed distance fields," *Proc. Conf. Robot. Sci. Syst.*, vol. 4, no. 1, pp. 1–9, 2015.

[30] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5556–5565.

[31] O. Kähler et al., "Very high frame rate volumetric integration of depth images on mobile devices," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 11, pp. 1241–1250, Nov. 2015.

[32] W. Cheng, L. Xu, L. Han, Y. Guo, and L. Fang, "iHuman3D: Intelligent human body 3D reconstruction using a single flying camera," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 1733–1741.
[33] D. Zhong, L. Han, and L. Fang, "idFusion: Globally consistent dense 3D reconstruction from RGB-D and inertial measurements," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 962–970.
[34] H. Pfister, M. Zwicker, J. Van Baar, and M. Gross, "Surfels: Surface elements as rendering primitives," in *Proc. 27th Annu. Conf. Comput. Graph. Interactive Techn.*, 2000, pp. 335–342.
[35] J. Stückler and S. Behnke, "Multi-resolution surfel maps for efficient dense 3D modeling and tracking," *J. Vis. Commun. Image Representation*, vol. 25, no. 1, pp. 137–147, 2014.
[36] T. Weise, T. Wismer, B. Leibe, and L. Van Gool, "Online loop closure for real-time interactive 3D scanning," *Comput. Vis. Image Understanding*, vol. 115, no. 5, pp. 635–648, 2011.
[37] D. Lefloch, M. Kluge, H. Sarbolandi, T. Weyrich, and A. Kolb, "Comprehensive use of curvature for robust and accurate online surface reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2349–2365, Dec. 2017.
[38] W. Gao and R. Tedrake, "Surfelwarp: Efficient non-volumetric single view dynamic reconstruction," in *Proc. Conf. Robot. Sci. Syst.*, 2018, pp. 1–10.
[39] Z. Yan, M. Ye, and L. Ren, "Dense visual SLAM with probabilistic surfel map," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 11, pp. 2389–2398, Nov. 2017.
[40] D. Lefloch, T. Weyrich, and A. Kolb, "Anisotropic point-based fusion," in *Proc. 18th Int. Conf. Inf. Fusion*, 2015, pp. 2121–2128.
[41] W. Yifan, F. Serena, S. Wu, C. Öztireli, and O. Sorkine-Hornung, "Differentiable surface splatting for point-based geometry processing," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–14, 2019.
[42] D. Chen, Z. Tang, Z. Xu, Y. Zheng, and Y. Liu, "Gaussian fusion: Accurate 3D reconstruction via geometry-guided displacement interpolation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 5916–5925.
[43] Y. Xu, L. Nan, L. Zhou, J. Wang, and C. C. Wang, "HRBF-fusion: Accurate 3D reconstruction from RGB-D data using on-the-fly implicits," *ACM Trans. Graph.*, vol. 41, no. 3, pp. 1–19, 2022.
[44] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–14, 2023.
[45] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2D Gaussian splatting for geometrically accurate radiance fields," in *Proc. SIGGRAPH 2024 Conf. Papers*, 2024, pp. 1–11.
[46] P. Wu, Y. Liu, M. Ye, J. Li, and S. Du, "Fast and adaptive 3D reconstruction with extensively high completeness," *IEEE Trans. Multimedia*, vol. 19, pp. 266–278, 2017.
[47] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–13, 2013.
[48] M. Kazhdan, M. Chuang, S. Rusinkiewicz, and H. Hoppe, "Poisson surface reconstruction with envelope constraints," *Comput. Graph. Forum*, vol. 39, no. 5, pp. 173–182, 2020.
[49] M. Kazhdan and H. Hoppe, "Distributed poisson surface reconstruction," *Comput. Graph. Forum*, vol. 42, no. 6, 2023, Art. no. e14925.
[50] M. Kohlbrenner, S. Lee, M. Alexa, and M. Kazhdan, "Poisson manifold reconstruction—Beyond co-dimension one," *Comput. Graph. Forum*, vol. 42, no. 5, 2023, Art. no. e14907.
[51] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust," in *Proc. 6th ACM Symp. Solid Model. Appl.*, 2001, pp. 249–266.
[52] D. Cohen-Steiner and F. Da, "A greedy delaunay-based surface reconstruction algorithm," *Vis. Comput.*, vol. 20, pp. 4–16, 2004.
[53] T. K. Dey and S. Goswami, "Provable surface reconstruction from noisy samples," in *Proc. 20th Annu. Symp. Comput. Geometry*, 2004, pp. 330–339.
[54] T. K. Dey and L. Wang, "Voronoi-based feature curves extraction for sampled singular surfaces," *Comput. Graph.*, vol. 37, no. 6, pp. 659–668, 2013.
[55] A. D. Parakkat, S. Ohrhallinger, E. Eisemann, and P. Memari, "Ballmerge: High-quality fast surface reconstruction via voronoi balls," *Comput. Graph. Forum*, vol. 43, no. 2, 2024, Art. no. e15019.
[56] M. Berger et al., "State of the art in surface reconstruction from point clouds," in *Proc. 35th Annu. Conf. Eur. Assoc. Comput. Graph.*, 2014, pp. 161–185.
[57] C. Shen, J. F. O'Brien, and J. R. Shewchuk, "Interpolating and approximating implicit surfaces from polygon soup," in *Proc. ACM SIGGRAPH Papers*, 2004, pp. 896–904.
[58] Z.-Q. Cheng et al., "A survey of methods for moving least squares surfaces," in *Proc. ACM Conf. vol. Graph. Point-Based Graph. SIGGRAPH*, 2008, pp. 9–23.
[59] R. Kolluri, "Provably good moving least squares," *ACM Trans. Algorithms*, vol. 4, no. 2, pp. 1–25, 2008.
[60] M. Alexa et al., "Point set surfaces," in *Proc. Vis.*, 2001, pp. 21–29.
[61] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized GICP for fast and accurate 3D point cloud registration," in *Proc. IEEE Conf. Robot. Automat.*, pp. 11054–11059, 2021.
[62] A. C. Öztireli, G. Guennebaud, and M. Gross, "Feature preserving point set surfaces based on non-linear kernel regression," in *Proc. Conf. Comput. Graph. Forum*, vol. 28, no. 2, pp. 493–501, 2009.
[63] J. Wang, M. Xu, F. Foroughi, D. Dai, and Z. Chen, "FasterGICP: Acceptance-rejection sampling based 3D lidar odometry," *IEEE Robot. Automat. Lett.*, vol. 7, no. 1, pp. 255–262, Jan. 2022.
[64] J. Straub et al., "The replica dataset: A digital replica of indoor spaces," 2019, *arXiv:1906.05797*.
[65] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 224–236.
[66] K.-C. Chan, J. Xiao, H. L. Goshu, and K.-M. Lam, "Point cloud densification for 3D Gaussian splatting from sparse input views," in *Proc. 32nd ACM Int. Conf. Multimedia*, 2024, pp. 896–904.

Duo Chen received the Ph.D. degree in computer science and technology from Sichuan University, Chengdu, China, in 2022. He is currently with the School of Artificial Intelligence, Chongqing University of Education, Chongqing, China, and a Postdoctoral Researcher with the MOE Key Lab for NeuroInformation, School of Life Science and Technology, Clinical Hospital of Chengdu Brain Science Institute, University of Electronic Science and Technology of China, Chengdu, and also with the Chongqing University Industrial Technology Research Institute, Chongqing. He has authored or coauthored for many top-tier academic journals and conferences, including IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, Optics Letters, ICCV, and ACM MM. His main research interests include computational imaging, multimedia, and computer vision with a particular focus on 3D reconstruction. He was a reviewer for many top-tier academic journals and conferences, including IEEE TRANSACTIONS ON MULTIMEDIA, CVPR, ICCV, ICML, AAAI, and ACM MM.

Zixin Tang received the M.S. degree from the China Academy of Ordnance Science, Beijing, China, in 2016, and the Ph.D. degree in computer science from Sichuan University, Chengdu, China, in 2023. He is currently a Lecturer with the School of Computing and Artificial Intelligence, Southwest University of Finance and Economics, Chengdu. His research interests include image restoration, computational imaging, and machine learning.

Ke Song received the M.S. and Ph.D. degrees from Sichuan University, Chengdu, China, in 2006 and 2020, respectively. He is currently an Associate Professor with the Chongqing University of Education, Chongqing, China. His research interests include adaptive beamforming and signal processing.

Xingyu Peng is currently working toward the B.S. degree in artificial intelligence from the School of Artificial Intelligence, Chongqing University of Education, Chongqing, China. He is under the supervision of Dr. Duo Chen with Engineering Practice Project in the Excellent Engineer Class.

Wuque Cai received the B.S. degree in mechanical engineering from the College of Mechanical and Electrical Engineering, Hohai University, Nanjing, China, in 2018. He is currently working toward the Ph.D. degree in biomedical engineering with the University of Electronic Science and Technology of China, Chengdu, China. His main research interests include computer vision and spiking neural networks, with a particular focus on gesture recognition.

Hongze Sun received the B.S. degree in intelligent science and technology from the School of Artificial Intelligence, Xidian University, Xi'an, China, in 2019. He is currently working toward the Ph.D. degree in biomedical engineering with the School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, China. His main research interests include brain-inspired intelligence, deep learning, and computer vision.

Dezhong Yao (Senior Member, IEEE) received the Ph.D. degree in applied geophysics from the Chengdu University of Technology, Chengdu, China, in 1991, and the second Ph.D. degree in biomedical science from Aalborg University, Aalborg, Denmark, in 2005. He is currently a Full Professor of neuroengineering and neurodata with the University of Electronic Science and Technology of China (UESTC), Chengdu. He is also the Dean of the Sichuan Institute for Brain Science and Brain-Inspired Intelligence, Chengdu. Since 1991, he has been with the UESTC as a Postdoctoral Researcher in 1991, Associate Professor in 1992, Full Professor in 1995, and a Changjiang Scholar Professor in 2006. He founded the School of Life Science and Technology, UESTC, in 2001, and the Ministry of Education (MOE) Key Laboratory for NeuroInformation, UESTC, in 2009. His research interests include EEG, simultaneous EEG, and functional magnetic resonance imaging, and brain–apparatus communication. Dr. Yao was a fellow of the American Institute for Medical and Biological Engineering in 2017. He was the recipient of the Roy John Award in 2018. He has been the Vice President of the Chinese Society of Biomedical Engineering since 2014 and the Chairperson of the Chinese EEG Consortium since 2019. He is the Chief Editor of Brain-Apparatus Communication and an Associate Editor for IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING.

Daqing Guo received the B.S. degree in automation and computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2004, the M.S. degree in computer science from the Graduate University of the Chinese Academy of Sciences, Beijing, China, in 2007, and the Ph.D. degree in circuits and systems from the University of Electronic Science and Technology of China in 2011. From 2011 to 2012, he was a Postdoctoral Researcher with Computational Neuroscience Unit, Okinawa Institute of Science and Technology Graduate University, Okinawa, Japan. He is currently a Full Professor with the Clinical Hospital of Chengdu Brain Science Institute, Ministry of Education (MOE) Key Laboratory for NeuroInformation, School of Life Science and Technology, University of Electronic Science and Technology of China. He has authored or coauthored more than 80 publications for various journals and conferences. His current research interests include computational neuroscience, brain-inspired intelligence, and digital twin brains.