HRBF-Fusion: Accurate 3D Reconstruction from RGB-D Data Using On-the-Fly Implicits

YABIN XU, Nanjing University of Aeronautics and Astronautics, China / Delft University of Technology, The Netherlands
LIANGLIANG NAN, Delft University of Technology, The Netherlands
LAISHUI ZHOU, Nanjing University of Aeronautics and Astronautics, China
JUN WANG, Nanjing University of Aeronautics and Astronautics, China
CHARLIE C.L. WANG, The University of Manchester, United Kingdom / Delft University of Technology, The Netherlands

Fig. 1. Reconstruction of two large indoor scenes: (left) a study room of a university library and (right) a study platform in a grand hall of an academic building. The original two sequences consists of 16,128 (library) and 10,930 (study platform) RGB-D image frames and the reconstructed model consists of 7,488,867 (library) and 7,904,727 (study platform) points respectively. The average processing speed of our approach is around 43ms per frame, which demonstrates a nearly real-time performance. RGB-D data in these two experiments are captured by a Microsoft Kinect v1 sensor with a resolution of 640 × 480. Progressive results of the reconstruction can be found in the supplementary video.

Reconstruction of high-fidelity 3D objects or scenes is a fundamental research problem. Recent advances in RGB-D fusion have demonstrated the potential of producing 3D models from consumer-level RGB-D cameras. However, due to the discrete nature and limited resolution of their surface representations (e.g., point- or voxel-based), existing approaches suffer from the accumulation of errors in camera tracking and distortion in the reconstruction, which leads to an unsatisfactory 3D reconstruction. In this paper, we present a method using on-the-fly implicits of Hermite Radial Basis Functions (HRBFs) as a continuous surface representation for camera tracking in an existing RGB-D fusion framework. Furthermore, curvature estimation and confidence evaluation are coherently derived from the inherent surface properties of the on-the-fly HRBF implicits, which devote to a data fusion with better quality. We argue that our continuous but on-the-fly surface representation can effectively mitigate the impact of noise with its robustness and constrain the reconstruction with inherent surface smoothness when being compared with discrete representations. Experimental results on various real-world and synthetic datasets demonstrate that our HRBF-fusion outperforms the state-of-the-art approaches in terms of tracking robustness and reconstruction accuracy.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: 3D reconstruction, closed-form HRBFs, registration, camera tracking, fusion.

ACM Reference Format:
Yabin Xu, Liangliang Nan, Laishui Zhou, Jun Wang, and Charlie C.L. Wang. 2022. HRBF-Fusion: Accurate 3D Reconstruction from RGB-D Data Using On-the-Fly Implicits. *ACM Trans. Graph.* 1, 1, Article 1 (January 2022), 19 pages. https://doi.org/10.1145/3516521

1 INTRODUCTION

Reconstruction of high-fidelity 3D objects or scenes is vital to applications such as augmented / virtual reality, digital fabrication, and robotics. With the increasing popularity of consumer-level depth cameras (e.g., Microsoft Kinect), 3D information, in the form of RGB-D images or point clouds, can be easily obtained. A lot of reconstruction systems targeting on producing surface models of small-scale objects or large scenes [Cao et al. 2018; Choi et al. 2015; Dai et al. 2017; Keller et al. 2013; Lefloch et al. 2017; Whelan et al. 2016; Zhou and Koltun 2015] have been introduced since the pioneering work of *KinectFusion* [Newcombe et al. 2011]. Despite

Authors' addresses: Yabin Xu, yabinxu007@gmail.com, Nanjing University of Aeronautics and Astronautics, China / Delft University of Technology, The Netherlands; Liangliang Nan, Delft University of Technology, The Netherlands, liangliang.nan@gmail.com; Laishui Zhou, Nanjing University of Aeronautics and Astronautics, China, zlsme@nuaa.edu.cn; Jun Wang, Nanjing University of Aeronautics and Astronautics, China, wjun@nuaa.edu.cn; Charlie C.L. Wang, The University of Manchester, United Kingdom / Delft University of Technology, The Netherlands, changling.wang@manchester.ac.uk.

0730-0301/2022/1-ART1 $15.00
https://doi.org/10.1145/3516521

the advances in 3D reconstruction in the last decade, obtaining high-quality 3D models from consumer-grade depth cameras remains an open problem due to the following two main issues.

- *Imperfect Surface Representation*: Existing approaches lack an accurate surface representation that facilitates high-fidelity reconstruction while being memory efficient and computationally affordable. The volumetric representation is widely used for RGB-D reconstruction systems [Chen et al. 2013; Dai et al. 2017; Niessner et al. 2013] following *KinectFusion* [Newcombe et al. 2011]. However, a commonly used implementation with fixed-size resolution lacks adaptiveness [Chen et al. 2013; Dai et al. 2017; Niessner et al. 2013], which tends to generate over-smoothed surfaces in the regions with geometric details. The alternative surface representation [Keller et al. 2013] – surfel, which predicts geometry by ray-to-plane surfel splatting, works poorly in high-curvature regions and is also prone to failure due to noises and outliers.
- *Camera Tracking Error*: Imprecision registration based on Iterative Closest Point (ICP) or its variants [Besl and McKay 1992; Rusinkiewicz and Levoy 2001] is usually applied for camera pose estimation between RGB-D frames, where distortion errors are accumulated and can become significant in featureless regions. Research efforts have been paid to resolve the problem through global optimization [Choi et al. 2015; Zhou and Koltun 2013] or additional information provided by the RGB-D camera (e.g., geometric [Lefloch et al. 2017; Zhou and Koltun 2015] and photometric [Whelan et al. 2016] information), to derive a weighted variant of the ICP scheme to reduce camera tracking drift. In recent pipelines (e.g., [Cao et al. 2018; Dai et al. 2017; Whelan et al. 2016]), both strategies are applied to improve the result of reconstruction.

The issue of camera tracking is also suffered from the lack of good surface representation when geometric cues are employed to enhance ICP registration.

1.1 Our method

To address the aforementioned issues, we propose HRBF-Fusion, a new method using on-the-fly HRBF implicits for high-accurate camera tracking and high-fidelity 3D reconstruction. The core of our method is a voxel-free implicit surface representation, i.e., the closed-form HRBF surface approximation that gracefully benefits multiple key stages of the reconstruction pipeline, including preprocessing, camera pose estimation, and depth map fusion. The 3D reconstruction pipeline used in our tests is a variant of *ElasticFusion* [Whelan et al. 2016] and *ORB-SLAM2* [Mur-Artal and Tardós 2017], in which the tracking-and-fusion steps of ElasticFusion are used to generate submaps and the ORB-based local-to-global optimization routine is used to obtain a global consistent 3D model for large scenes. In contrast, we evaluate both the global model and the new RGB-D frames as continuous but compactly-supported HRBF surfaces to produce robust curvature estimation and reconstruction-indicated confidence maps. With the help of these HRBF surfaces, more reliable camera tracking and depth map fusion can be achieved. In summary, we make the following contributions:

- A method to evaluate a continuous surface effectively and efficiently on both the global model and the acquired RGB-D frame by using on-the-fly HRBF implicits;
- A robust and efficient curvature evaluation method based on the on-the-fly HRBF implicits, leading to a dramatic improvement in camera tracking based on the curvature-weighted registration;
- A reconstruction-indicated confidence evaluation method, also based on efficient HRBF surface evaluation, can significantly reduce the impact of noises and outliers in both camera tracking and depth-image fusion.

As a consequence, we develop a more robust reconstruction system for high-fidelity online surface reconstruction, which also shows good scalability to large scenes.

1.2 Related work

1.2.1 Geometric representation. 3D reconstruction within a commodity RGB-D camera has been extensively studied in the past decade. A key ingredient toward a high-quality 3D reconstruction system is the underlying representation for camera pose estimation and depth map fusion. Different representations have been proposed, including volumetric representation [Curless and Levoy 1996; Dai et al. 2017; Newcombe et al. 2011; Niessner et al. 2013; Zhang and Hu 2017], surfel-based representation [Cao et al. 2018; Keller et al. 2013; Weise et al. 2009; Whelan et al. 2016], height field [Meilland and Comport 2013], probability-based representation [Dong et al. 2018], and 2.5D depth map [Gallup et al. 2010]. A recently trend is to solve the problem by using neural implict representation for shape generation [Huang et al. 2021b,a; Liu et al. 2020; Sucar et al. 2021, 2020] and using learing-based method for depth fusion [Bozic et al. 2021; Weder et al. 2020, 2021]. Here we provide a compact solution by using a closed-form representation for the on-the-fly implicts.

Following the pioneering work of *KinectFusion* [Newcombe et al. 2011] that applied a Truncated Signed Distance Field (TSDF) [Curless and Levoy 1996] for modeling integration, volumetric representation has demonstrated promising results for reconstructing small-scale scenes. Because of its implementation on GPU for real-time tracking and fusion, volumetric representation becomes more and more popular [Chen et al. 2013; Dai et al. 2017; Meerits et al. 2018; Niessner et al. 2013]. The original uniform-grid KinectFusion has a fundamental limitation (i.e., the lack of scalability), which leads to expensive memory consumption for reconstructing fine details. Recently, a learning-based TSDF was adopted to represent the geometry under reconstruction [Sun et al. 2021]. Although methods have been developed to alleviate this by exploiting sparsity in the TSDF representation [Chen et al. 2013; Niessner et al. 2013], the quality of local reconstruction still depends on the resolution to partition the space which is related to the scale of the scene.

Kelly et al. [2013] proposed a surfel-based representation method to solve the scalability issue and has presented comparable results against volumetric methods on flat or smooth regions. In their method, a ray-to-plane surfel rendering algorithm is used to predict the model for real-time camera tracking. The method has been applied to real-time reconstruction systems [Cao et al. 2018; Lefloch et al. 2017; Whelan et al. 2016]. However, the linear ray-to-plane based shape prediction is sensitive to noises in particular on the

high-curvature surface regions. Hence reconstructed models are often distorted when there are noises in high-curvature regions. Implicit moving least-squares (IMLS) surface was employed in [Liu et al. 2021] to achieve a better shape representation in their learning-based 3D reconstruction. However, the evaluation of IMLS is less efficient. Differently, we predict the shape from surfels by using closed-form HRBF implicits which makes our system is memory-efficient and robust.

Radial Basis Functions was employed in [Carr et al. 2001] for surface reconstruction. In this method, the computation is however very time-consuming, and it also requires the provision of auxiliary 'off-surface' points. Liu et al. [2016] introduced closed-form HRBF implicits using quasi-interpolation, which has demonstrated its capability of generating surface reconstruction in high quality and high efficiency. Inspired by this work, we explore the possibility to incorporate the closed-form HRBF implicits with the inherent surface properties for noise-resistant camera tracking and high-quality 3D reconstruction. It is also worthy to notice that Schöps et al. [2020] recently developed an online mesh construction method for reconstruction refinement; nevertheless, camera poses are required as additional input for their method. Differently, our on-the-fly HRBF implicits are directly devoted to camera tracking and RGB-D reconstruction.

1.2.2 Camera tracking. An important issue in the real-time RGB-D surface reconstruction system is the drift of camera tracking caused by the instability of the frame-to-model registration.

One of the reasons that cause the instability of registration is the presence of noise and outliers. To mitigate the impact of noise and outliers, Jian and Vemuri [2011] proposed to use the Gaussian Mixture Model (GMM) to describe the distribution of both template and point set. Not only geometric but also color information has been conducted for probabilistic registration [Danelljan et al. 2016]. Although robust, these probabilistic registration approaches are time-consuming which makes them ineligible for real-time reconstruction from a sequence of input RGB-D frames. Others tend to evaluate the reliability of an input raw depth map by analyzing the inherent property of depth cameras (e.g., [Reynolds et al. 2011]). Similarly, a distortion-based model is employed in [Keller et al. 2013] which weights measurements based on the assumption that the depth data captured near the center of a sensor are more accurate. Recently, a voting mechanism is introduced in [Cao et al. 2018] to evaluate the confidence of depth map for generalized ICP [Segal et al. 2009] by using the time-coherence between nearby frames. In this paper, we propose a novel reconstruction-indicated confidence metric to exploit the underlying uncertainty on each depth map.

Another reason for tracking drift is the lack of salient geometric features in the scene which leads to slippery registration. As depth cameras are commonly equipped with an additional RGB camera, colors are used as additional information to form a joint optimization problem [Godin et al. 1994; Whelan et al. 2016] or to pre-align the depth map with color-based features [Henry et al. 2012]. Yang et al. [2017] incorporated visual saliency into a volumetric fusion pipeline to achieve high-quality object reconstruction. Other geometric features have also been considered in other approaches to add weights in the optimization for registration, including contour cues [Zhou and Koltun 2015], planar structures, and repeated objects [Zhang et al. 2015], patch co-planarity [Shi et al. 2018] and curvatures [Lefloch et al. 2017]. Among them, the curvature is very general and can be evaluated in all regions. Several methods of curvature estimation have been discussed in [Lefloch et al. 2017], among which the method of adjacent-normal cubic approximation [Goldfeather and Interrante 2004] is concluded as the most robust curvature estimator. However, this method needs to solve a 7×7 linear system at every point, which hinders its usage for real-time applications even with the implementation on GPU (ref. [Lefloch et al. 2017]). By the requirement of real-time performance, Lefloch et al. [2017] selected the chord-and-normal-vectors (CAN) approach [Zhang et al. 2008] for curvature estimation. The camera drift can be reduced by integrating curvature information into the ICP framework with higher weights in high curvature regions [Lefloch et al. 2017]. However, the curvature evaluation in their approach is not robust when input RGB-D data becomes noisy. This is crucial as the input frames from consumer-level RGB-D cameras are often contaminated with noises and outliers. In our approach, we use curvature as additional information in both tracking and fusion stages – but differently, curvature in our approach is robustly extracted from the continuous surfaces represented by HRBF implicit.

1.2.3 Accumulated error. Apart from focusing on the error sourced from the frame-by-frame registration, methods have been developed to correct the error accumulation in camera pose estimation and global 3D model in both online [Cao et al. 2018; Dai et al. 2017; Wang and Guo 2017; Wasenmüller et al. 2016; Whelan et al. 2012] and offline [Choi et al. 2015; Li et al. 2013; Zhou and Koltun 2013; Zhou et al. 2013] mode, where the offline methods are time-consuming.

For online correction, Whelan et al. [2016] proposed a system that divides the reconstructed model into active (recently captured frames) and inactive parts. When the registration between active and inactive parts is successful, an optimization-based deformation is applied to deform the active part to fuse into the inactive part. However, the routine does not provide a way to fix the errors that have already been inherited into the inactive part. Yang et al. [2020] proposed a noise-resilient panoramic scanning approach that uses robot-mounted multiple RGB-D cameras to obtain high-quality 3D models of the scene. A different strategy is applied in the area of simultaneous localization and mapping (SLAM) [Engel et al. 2013; Forster et al. 2014; Klein and Murray 2007; Mur-Artal et al. 2015; Mur-Artal and Tardós 2017], where drift-free pose estimation has been extensively studied. The basic idea of these approaches is to minimize the reprojection error across frames or distribute camera pose estimation error across the pose-graph constructed by the co-visibility between frames. While focusing on different problems, these approaches do not provide a method to correct dense 3D models generated from depth map fusion. To solve this problem, submap-based online reconstruction systems (e.g., [Cao et al. 2018; Dai et al. 2017]) are proposed to correct the camera poses and minimize the geometric error of 3D models in an integrated manner. In our system, we adopt a similar submap-based hierarchical optimization for the steps of close-loop detection, camera pose, and 3D model correction.

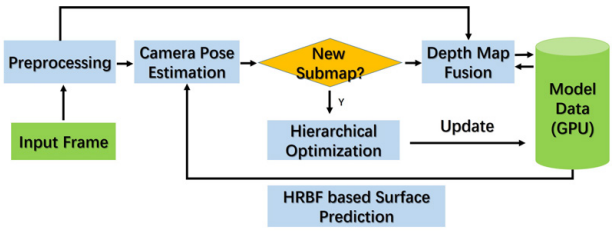


Fig. 2. Framework of the proposed RGB-D reconstruction system.

2 OVERVIEW

We utilize closed-form HRBF implicits for on-the-fly surface evaluation for the global model, which replaces the commonly used discrete surface representations of existing reconstruction systems and plays a vital role in the key stages of the pipeline to improve tracking robustness and reconstruction accuracy. We adopt a computational framework similar to prior systems [Dai et al. 2017; Keller et al. 2013; Lefloch et al. 2017; Newcombe et al. 2011; Niessner et al. 2013] for reconstruction (see Fig. 2). The functionality of closed-form HRBF implicits is utilized in various stages of the framework.

The global model $\mathbf{M}$ is represented by a set of unorganized points where each point is associated with attributes[1] including its position $\bar{\mathbf{v}} \in \mathbb{R}^3$, normal $\bar{\mathbf{n}} \in \mathbb{R}^3$, support size $\bar{r} \in \mathbb{R}$, confidence value $\bar{c} \in \mathbb{R}$, and two principal curvature values $\bar{\kappa}_1$, $\bar{\kappa}_2 \in \mathbb{R}$. This is a highly scalable representation, which can be considered as an enriched surfel representation [Cao et al. 2018; Keller et al. 2013].

When capturing a new RGB-D frame $\mathbf{F} = \{\mathbf{D}, \mathbf{C}\}$ with $\mathbf{D}$ and $\mathbf{C}$ denoting the depth map and the color map respectively, the RGB-D frame $\mathbf{F}$ is fused into the global model by applying the following key steps:

- *Preprocessing*: Continuous surfaces are evaluated in the input RGB-D frame and on the global model by using the on-the-fly HRBF implicits respectively (Section 3.1). Note that the HRBF surface for the global model is evaluated in the previous frame of the scanning sequence. With the help of robust HRBF surface evaluation, a curvature map (Section 3.2) and a reconstruction-indicated confidence map (Section 3.3) are evaluated in the input frame to enhance the robustness of our reconstruction pipeline.
- *Camera pose estimation*: The purpose of this step is to obtain the transformation between the input frame and the current global model. We adopt a variant ICP algorithm based on the point-to-plane metric with specially designed searching and weighting schemes to align it to the surface predicted from its last pose. Unlike existing RGB-D reconstruction systems based on discrete surface representations, our accurate and robust local surface reconstruction based on HRBF implicits improves the robustness in both the correspondence search (Section 4.1) and the optimization of registration (Section 4.2). On-the-fly calculated curvatures and normals are stored in local but 'dense' maps for camera pose estimation, which can avoid the problem caused by sparsity in a global map.
- *Depth map fusion*: To integrate a new frame into the global model with a valid pose, correspondences between vertices of the input frame and the points in the global model are established based on an index map that is obtained by rendering the index of each model point into a texture [Keller et al. 2013]. After that, the input vertices with their attributes are merged into the global model using a confidence-weighted average (Section 5). Similar to other surfel-based approaches (e.g., [Cao et al. 2018; Keller et al. 2013]), attributes stored on the global model are employed to conduct the fusion.

These steps are repeated until the relative translation between the first frame and the current frame exceeds a certain threshold. Then, the global model formed by already registered and fused frames will be treated as a submap.

With reliable geometric and photometric enhanced registration, high-quality camera tracking and surface reconstruction can be achieved for relatively small objects. When reconstructing large scenes by long-range scanning, a local-to-global optimization scheme similar to [Cao et al. 2018] is applied between submaps to further alleviate the accumulation of errors in camera tracking by using the ORB features [Rublee et al. 2011].

3 GEOMETRIC CUES BY HRBF IMPLICITS

In this section, we first introduce the method of surface prediction with closed-form HRBF implicits. After that, the robust curvatures and the reconstruction-indicated confidence map can be generated from the on-the-fly HRBF surfaces.

HRBF implicits have been used to reconstruct an implicit function from scattered Hermite points [Macêdo et al. 2011]. Given a point set $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, ...\mathbf{p}_n\}$ with corresponding normals $\mathbf{N} = \{\mathbf{n}_1, \mathbf{n}_2, ...\mathbf{n}_n\}$, a function f interpolating the positions and the normals can be defined as

$$f(\mathbf{x}) = \sum_{j=1}^{n} \{\alpha_j \psi(\mathbf{x} - \mathbf{p}_j) - < \boldsymbol{\beta}_j, \nabla\psi(\mathbf{x} - \mathbf{p}_j) >\}, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot-product of two vectors, and ∇ is the gradient operator. The *Compactly Supported Radial Basis Functions* (CSRBF) [Wendland 1995] are applied as the kernels because of their numerical stability and the on-the-fly nature. Specifically, we have

$$\psi(\mathbf{x} - \mathbf{p}_j) = \begin{cases} (1 - \frac{d}{r})^4 (\frac{4d}{r} + 1), & d \in [0, r], \\ 0, & otherwise, \end{cases} \tag{2}$$

where $d = \|\mathbf{x} - \mathbf{p}_j\|$ is the Euclidean distance between the query point and the corresponding CSRBF kernel and r is the support size. The coefficients $\alpha_j \in \mathbb{R}$ and $\boldsymbol{\beta}_j \in \mathbb{R}^3$ can be computed from the following constraints: $f(\mathbf{p}_i) = 0$ and $\nabla f(\mathbf{p}_i) = \mathbf{n}_i$ on all given points $\mathbf{p}_{i=1,...,n}$. Instead of solving a $4n \times 4n$ linear system, a closed-form function was proposed in [Liu et al. 2016] to approximate the HRBF implicits as

$$\hat{f}(\mathbf{x}) = -\sum_{j=1}^{n} < \frac{r_j^2}{20 + \eta r_j^2} \mathbf{n}_j, \nabla\psi(\mathbf{x} - \mathbf{p}_j) >, \tag{3}$$

where r_j is the support size of kernel centered at $\mathbf{p}_j$. The value of r_j should be determined to cover at least 8 neighboring kernels for constructing a locally continuous surface [Liu et al. 2016] around

[1] Variables evaluated on the global model are represented by symbols with '¯' head throughout the paper.

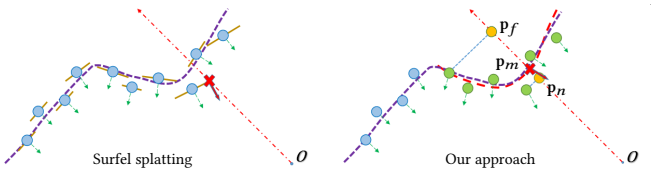


Fig. 3. An illustration of the surfel splatting (left) [Cao et al. 2018; Keller et al. 2013] and our HRBF-based (right) surface prediction methods. The red cross in each figure represents the intersection between the ray (red dashed line) and the global model points (blue dots).

$\mathbf{p}_j$. $\eta = 1.0 \times 10^6$ is employed as the regularization coefficient for points evaluated in the unit of meter [Liu et al. 2016]. With such a closed-form surface representation, solving the linear system can be avoided. This enables a method for efficient and on-the-fly surface evaluation, which is very important for real-time reconstruction.

3.1 Surface evaluation

In our approach, continuous surfaces are evaluated for both the newly captured depth image and the global model by using the on-the-fly HRBF implicits. Specifically, two surfaces are evaluated on all pixels of two frames – the input RGB-D frame for a local model and the previous frame in the scanning sequence for the global model. Similar to the raycasting method of [Newcombe et al. 2011], we predict the surface points for a pixel $\mathbf{u}$ at the current pose by intersecting the HRBF local surface with the ray from the camera optical center to the corresponding point in the image plane (see Fig. 3). In contrast to the popular surfel-based surface prediction method [Cao et al. 2018; Keller et al. 2013] that searches for the nearest (from the viewpoint) discrete point within a radius (see the left of Fig. 3 for illustration), our method takes advantage of the smooth nature of the surface and thus is more robust to noise and outliers.

For the surface evaluation in a frame by HRBF implicits, we choose the kernels that are closer to the viewpoint while discarding kernels that have greater depth deviation from the nearest model point due to depth discrepancy. After obtaining a local set of kernels that define the HRBF surface on a viewing ray, we project the kernels' centers onto the ray to form a searching interval $[\mathbf{p}_n, \mathbf{p}_f]$, where $\mathbf{p}_n$ is the nearest point and $\mathbf{p}_f$ is the furthest one along the viewing ray. The model point $\mathbf{p}_m$ is supposed to lie in the interval to satisfy $\hat{f}(\mathbf{p}_m) = 0$, which can be obtained by a binary searching algorithm (see the right of Fig. 3 for an illustration). After determining the position of a surface point, other attributes at $\mathbf{p}_m$ such as colors can be predicted from its nearest kernel. Note that, this ray-intersection based surface evaluation can run in highly parallel mode on the many cores of GPU. Specifically, we implement the surface evaluation of HRBF implicits in a fragment shader that is used for per-pixel operation with each viewing ray defined on pixels. The input vertex map and normal map are bound with the fragment shader for local searching. The HRBF implicits are constructed and evaluated within the fragment shader. The outputs are the texture maps bound with a frame buffer, which are the predicted surface points and their corresponding attributes.

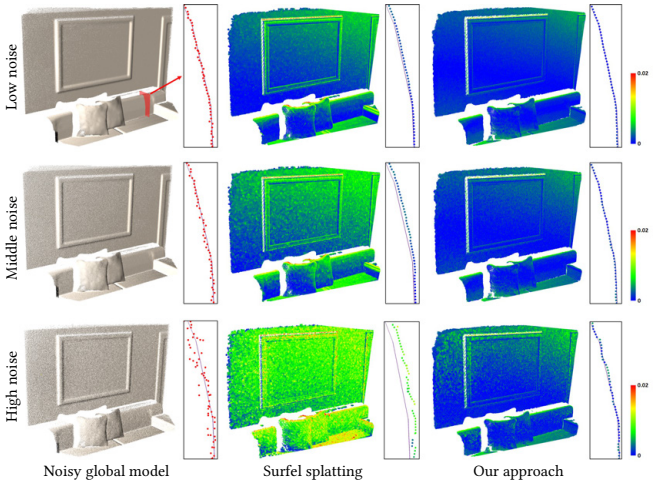


Fig. 4. Comparison of the predicted vertex map generated by surfel splatting (middle column) and HRBF local surface reconstruction (right column) on the same global model (left column). The test is conducted on the *lr kt1* example from *ICL-NUIM* [Handa et al. 2014] by adding three levels of Gaussian noise with the standard deviations as $\sigma = 3.0$, $\sigma = 6.0$ and $\sigma = 12.0$ respectively, where the ground truth of the geometry and camera poses are provided. Note that the global models are generated by fusing multiple RGB-D frames (i.e. 1-68) using the same strategy of [Keller et al. 2013] and the ground-truth camera poses. Colors indicate the unsigned distances from the points to the ground-truth 3D model.

For surface evaluation on a global model $\mathbf{M}$, the stored points $\{\bar{\mathbf{v}}\}$ will be used as the kernels of HRBF implicits. The resultant intersection points are stored in a 3D vertex map $\bar{\mathbf{V}}$. The normal at each intersection point $\mathbf{p}_m$ can also be obtained from the gradient as $\nabla\hat{f}(\mathbf{p}_m)/\|\nabla\hat{f}(\mathbf{p}_m)\|$. The resultant normal map is denoted by $\bar{\mathbf{N}}$. With the help of the closed-form HRBF implicits, we are able to predict $\bar{\mathbf{V}}$ more accurately – see the comparison with surfel splatting on a model with ground-truth geometry (Fig. 4). The experiment is conducted on the *lr kt1* example from the synthetic dataset *ICL-NUIM* [Handa et al. 2014] with the ground-truth geometry and camera poses provided. To evaluate the sensitivity to noise, the input RGB-D frames are contaminated by adding different levels of Gaussian noise. The global models are obtained by fusing multiple (i.e, 1-68) input frames with the ground-truth camera poses, while the same strategy of [Keller et al. 2013] is adopted for depth map fusion. As can be observed from the cross-sectional views in Fig. 4, the increased level of noise makes the points in the global model corrupt gradually. The surfel splatting method results in imprecise prediction of the underlying surface when highly noisy input is given. In contrast, the vertex map predicted by our method can properly represent the underlying surface. Moreover, smoother normal maps can be generated by our method (Fig. 5). Note that an accurate and robust prediction of geometry is the key ingredient to the high accuracy in camera pose estimation (Section 4). With our HRBF-based local surface reconstruction, the accuracy of geometry prediction and thus the registration is dramatically improved (see Fig. 6 for an example).

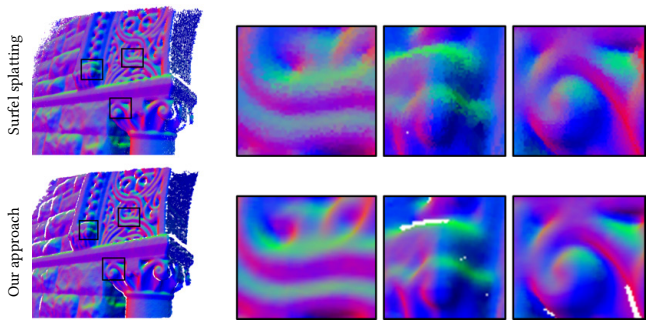


Fig. 5. Comparison of the normal maps generated by the ray-to-plane surfel splatting method (top row) and our HRBF-based prediction method (bottom row) on the *stone wall* from 3D Scene Data [Zhou and Koltun 2015].

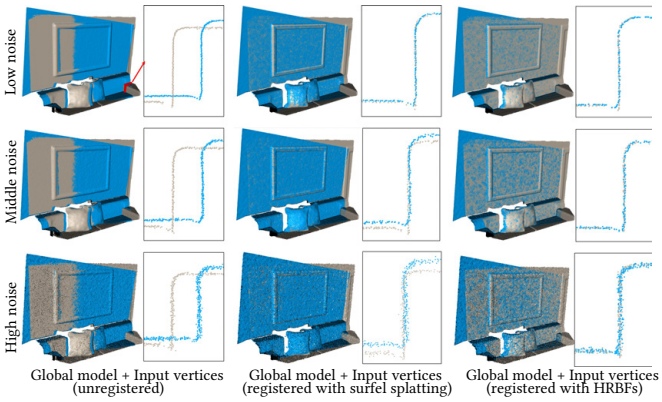


Fig. 6. Comparison of registration between the global model and the input vertices under different noise levels. Similar to already discussed in Fig. 4, the global model is obtained by fusing multiple RGB-D frames (i.e. 1-68) using the same strategy of [Keller et al. 2013] and the ground-truth camera poses. The left column shows the initial alignment of the global model and the input vertices, whereas the other two columns show registration results based on surfel splatting (the middle column) and our HRBF based method (the right column).

The kernels for surface evaluation in the input RGB-D frame are determined differently. Preprocessing is needed before applying the HRBF based surface evaluation. Given an input frame with the depth map and the color map, its corresponding 3D vertex map $\mathbf{V}$ is computed using the camera intrinsic matrix $\mathbf{K}$ by following the same steps as KinectFusion [Newcombe et al. 2011]. After applying a bilateral filter to reduce noise while preserving discontinuity in the depth map $\mathbf{D}$, the corresponding 3D vertex for each pixel $\mathbf{u} = (x, y)^T \in \mathbb{R}^2$ is computed as $\mathbf{V}(\mathbf{u}) = \mathbf{D}(\mathbf{u})\mathbf{K}^{-1}(\mathbf{u}^\top, 1.0)^T$. The corresponding normal map $\mathbf{N}$ can be derived from $\mathbf{V}$ by central difference. Besides, we assign each vertex with a support size $\mathbf{S}(\mathbf{u})$ for local HRBF surface evaluation. To construct a continuous HRBF surface, the support size of a kernel should cover at least k other kernels (i.e., $k = 8$ according to [Liu et al. 2016]). The k-nearest neighbor for each pixel $\mathbf{u}$ is first obtained by searching the vicinity of $\mathbf{u}$ in a window patch (i.e., 7×7) of the filtered vertex map $\mathbf{D}$.

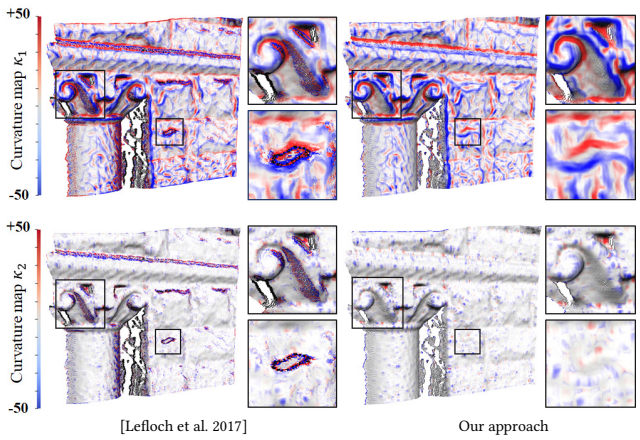


Fig. 7. Comparison of principal curvature estimated by [Lefloch et al. 2017] versus our method. The black points indicate the corresponding curvature values are out of a range of $[-300, 300]$. Note that, $|\kappa_1|, |\kappa_2| > 300$ means the radius of curvature is already less than $3mm$. These are geometric details that cannot be captured by RGB-D cameras – i.e., unreliable estimation.

The support size is assigned as the distance between a kernel and its k-th nearest neighbor. Lastly, the ray-intersection based surface evaluation is conducted in the input RGB-D frame to update its vertex map $\mathbf{D}$ and normal map $\mathbf{N}$.

3.2 Robust curvature

The principal curvature map κ is evaluated by the on-the-fly HRBF implicits in an input RGB-D frame, which provides important clues in the registration step (Section 4.2). Benefit from the continuous surface representation provided by HRBF implicit, the mean curvature H and the Gaussian curvature G can be reliably computed by the gradient and Hessian matrix of the function $\hat{f}(\cdot)$.

$$H = \frac{\nabla\hat{f}\mathbf{Hess}(\hat{f})\nabla\hat{f}^T - |\nabla\hat{f}|^2 Trace(\mathbf{Hess}(\hat{f}))}{2|\nabla\hat{f}|^3},$$
$$G = \frac{\begin{vmatrix}\mathbf{Hess}(\hat{f}) & \nabla\hat{f}^T \\ \nabla\hat{f} & 0\end{vmatrix}}{|\nabla\hat{f}|^4}, \tag{4}$$

where ∇ and $\mathbf{Hess}(\cdot)$ are the gradient and Hessian operator respectively. After that, the principal curvatures can be obtained by solving the quadratic equation of normal curvature derived constructed from the first and second fundamental forms [Patrikalakis 2002]. That is $\kappa_1 = H + \sqrt{H^2 - G}$ and $\kappa_2 = H - \sqrt{H^2 - G}$.

To evaluate the reliability of curvature estimation, a comparison between the prior approach [Lefloch et al. 2017] based on quadratic surface fitting and our method is given in Fig. 7. As can be observed in the zoom-view, curvature estimation applied by [Lefloch et al. 2017] is quite unstable in noisy regions (see the undefined points shown in black). In contrast, our approach based on local HRBF approximate is robust to noise. The result of curvature evaluation is stored in a map co-aligned with the vertex map $\mathbf{D}$.

3.3 Reconstruction-indicated confidence map

For each new input RGB-D frame, a confidence map is usually constructed to indicate the level of confidence at each vertex for the camera pose estimation and the depth fusion. In the previous reconstruction systems [Keller et al. 2013; Lefloch et al. 2017; Whelan et al. 2016], the confidence map Υ for each raw input is derived from the radial decreasing quality [Keller et al. 2013] according to the distortion model of the camera [Sarbolandi et al. 2015] – i.e., the depth values on pixels closer to the center of the camera are more accurate. The distortion-based method can improve the reconstruction quality to some extent but it still ignores the uncertainty of the input data. Hence we evaluate the input depth map by a reconstruction-indicated method based on the observation that the implicit surface reconstruction relies on the density and reliability of the acquired points.

The confidence is higher in regions where dense points exist to construct an implicit surface and vice versa [Wu et al. 2014]. Specifically, we evaluate the magnitude of the function gradient $\nabla\hat{f}(\mathbf{v})$ and its consistency to the normal $\tilde{\mathbf{n}}_D$ indicated by the depth map $\mathbf{D}$. This is because a reliable local shape described by HRBF implicits will 1) be commonly defined by more kernels and 2) have its gradient pointing toward the similar direction as $\tilde{\mathbf{n}}_D$. Therefore, the reconstruction-indicated confidence can be evaluated by

$$c_r = exp(-\frac{\varepsilon}{||\nabla\hat{f}(\mathbf{v}) \cdot \tilde{\mathbf{n}}_D||}) \tag{5}$$

with $\tilde{\mathbf{n}}_D$ being the unit normal obtained by applying central-difference on the bilateral filtered depth values of $\mathbf{D}$. ε is a coefficient to reflect the resolution of RGB-D cameras. For all our experimental tests taken on a Microsoft Kinect v1 sensor, $\varepsilon = 1000$ gives the best results. For each pixel $\mathbf{u}$, its final confidence is commonly determined by the reconstruction-indicated term c_r and the distortion-based term c_d as

$$c = c_r c_d, \tag{6}$$

where $c_d = exp(-\gamma^2/2\sigma^2)$ is the same as [Keller et al. 2013]. Here γ is the radial distance between the current pixel and the camera center normalized by the diagonal length of the frame image, and $\sigma = 0.6$ is derived empirically according to [Keller et al. 2013].

We compare our method of confidence map evaluation with the camera-distortion based method [Keller et al. 2013] on the *human* model from the CoRBS benchmark [Wasenmüller et al. 2016] (Fig. 8). In contrast to the method of Keller et al. [2013] that generates weights according to the optical direction of the camera, our method of confidence evaluation properly reflects the underlying uncertainty of the input frames (see the left column of Fig. 8 for an illustration). Moreover, we further evaluate the reconstruction results by using different confidence maps as shown in the right column of Fig. 8. As can be found in the zoom views, misalignment occurs by using the camera-distortion based method (see the double-layers in the zoom views of Fig. 8's top-right). Differently, our method can effectively reflect the unstable measurement in those regions with large depth variation by assigning smaller weight values. As a result, the registration based on our HRBF-based confidence evaluation provides better-aligned results (see the bottom-right of Fig. 8). We also measure the errors of reconstruction by the distance between each point to the ground-truth surface model, which are plotted in heat color with the corresponding histogram. In summary, our method leads to a more accurate 3D model with less artifact.

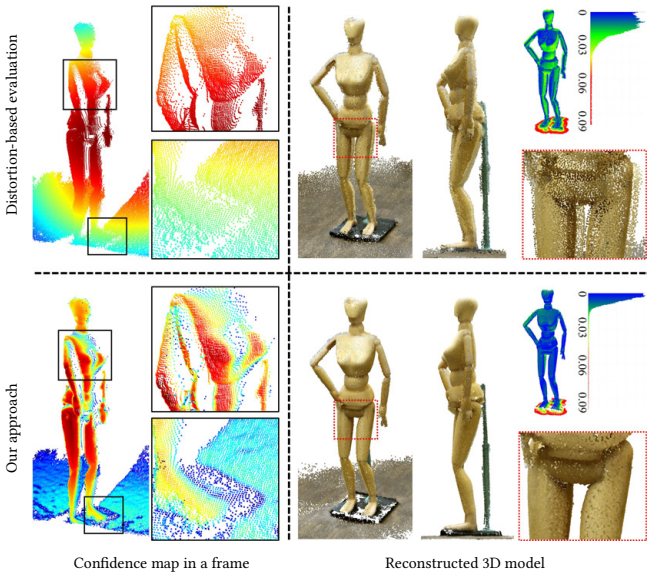


Fig. 8. Comparison of different methods for generating confidence maps in registration – the camera-distortion based [Keller et al. 2013] (top left) versus our reconstruction-indicated method (bottom left), where the test is conducted on the *human* model from the CoRBS benchmark [Wasenmüller et al. 2016]. The reconstructed models by using different confidence maps are shown in the right column, where zoom views highlight the quality difference in reconstruction. The reconstruction errors are measured by the distance between each point to the surface of the ground-truth model and are plotted in heat color with the corresponding histogram.

4 CAMERA POSE ESTIMATION

We estimate the camera pose of each newly captured RGB-D frame by registering it onto the global model, which highly depends on the underlying registration algorithm and is a key to 3D reconstruction in high accuracy. Our registration method consists of two steps:

(1) Searching the correspondence between each point of the input frame and its corresponding point in the vertex map predicted from the global model in the previous frame;
(2) Updating the registration transformation by minimizing the weighted point-to-plane geometric metric and the photometric difference between the pairs of points with correspondence determined in the first step.

These two steps are repeatedly applied until the registration converges to obtain the relative transformation between the neighboring frames. With the help of on-the-fly HRBF surfaces proposed in our approach, curvatures and confidence maps can be reliably estimated to improve the robustness of registration.

4.1 Correspondence search

Given a point $\mathbf{v}_i = \mathbf{V}_i(\mathbf{u})$ from the i-th frame (the input RGB-D frame), it is required to find its most similar point $\bar{\mathbf{v}}_{i-1}$ on the global model in the $(i-1)$-th frame's vertex map predicted by on-the-fly HRBF implicits. Assuming the motion between two consecutive frames is very small, the projective data association algorithm [Blais and Levine 1995] can be applied to speed up the search of correspondence (ref. [Keller et al. 2013; Newcombe et al. 2011]). Specifically, the estimated transformation $\mathbf{T}_i$ to the global model, which is initialized as $\mathbf{T}_{i-1}$ and will be updated during the iteration of registration, is used to transfer 3D points of the i-th frame into the previous frame by $\mathbf{T}_{i-1}^{-1}\mathbf{T}_i$. After that, we use a small window with a fixed size of 5×5 to search compatible points in the predicted vertex map of the global model.

We measure the dissimilarity of a point pair using the following metric similar to [Lefloch et al. 2017]

$$\gamma_d = \mu_d I_d + \mu_a I_a + \mu_c I_c, \tag{7}$$

which is determined by the distance variation term I_d, the angle variation term I_a and the curvature variation term I_c together with equal weight (i.e., $\mu_d = \mu_a = \mu_c = \frac{1}{3}$ works well in all our experiments).

$$\begin{aligned} I_d &= \|\mathbf{v}_i - \bar{\mathbf{v}}_m\|/R_{\max}, \\ I_a &= 1 - \mathbf{n}_i \cdot \bar{\mathbf{n}}_m/(\|\mathbf{n}_i\|\|\bar{\mathbf{n}}_m\|), \\ I_c &= 1 - exp\left(-\frac{|\kappa_{1,i} - \kappa_{1,m}| + |\kappa_{2,i} - \kappa_{2,m}|}{\max\{|\kappa_{1,m}|, |\kappa_{2,m}|\}}\right), \end{aligned} \tag{8}$$

where $\bar{\mathbf{v}}_m$, $\bar{\mathbf{n}}_m$, $\kappa_{1,m}$ and $\kappa_{2,m}$ are from the candidate points obtained from the global model. $R_{\max}$ is the distance between $\mathbf{v}_i$ and the farthest point that can be found in the search window.

A point pair with the smallest values of γ_d is considered as the valid corresponding points. Moreover, we apply a pruning strategy similar to [Newcombe et al. 2011] to discard outliers in the correspondence pairs. A reliable correspondence search depends on a robust normal and curvature estimation, which has been improved by using our on-the-fly HRBF surface evaluation. The pairs of compatible points are stored in a set $\Psi = \{(\mathbf{u}, \bar{\mathbf{u}})\}$ for computing the updated transformation $\mathbf{T}_i$.

4.2 Transformation update

The transformation is updated by minimizing an objective function considering both geometric and photometric information.

4.2.1 Geometric term. A curvature-based weight scheme [2017] is employed here to enhance the point-to-plane metric [Newcombe et al. 2011] for aligning an input RGB-D frame to the global model. The objective function to be minimized is defined as

$$E_{geom}(\mathbf{T}_i) = \sum_{(\mathbf{u},\bar{\mathbf{u}})\in\Psi} w(\bar{\mathbf{u}})\left((\mathbf{T}_i\mathbf{v}_i - \bar{\mathbf{v}}_{i-1}) \cdot \bar{\mathbf{n}}_{i-1}\right)^2, \tag{9}$$

where the curvature-based scheme [Lefloch et al. 2017] is employed to determine the weight $w(\bar{\mathbf{u}})$ by incorporating the confidence coefficient, the depth-value and most importantly the principal curvatures at the point $\bar{\mathbf{v}}_{i-1}(\bar{\mathbf{u}})$. Figure 9 demonstrates the performance improvement when using on-the-fly HRBF to evaluate curvatures (Section 3.2) and confidence map (Section 3.3) as proposed in this paper. In this experiment, the routine and the weighting scheme of registration are the same as [Lefloch et al. 2017]. We evaluate the camera tracking accuracy by computing Mean Camera Center Error and Standard Deviation (SD) between the estimated poses with the corresponding reference poses, as described in [Lefloch et al. 2017]. This comparison indicates that our method significantly improves the accuracy of registration (therefore camera pose estimation) when high-level noise is presented.

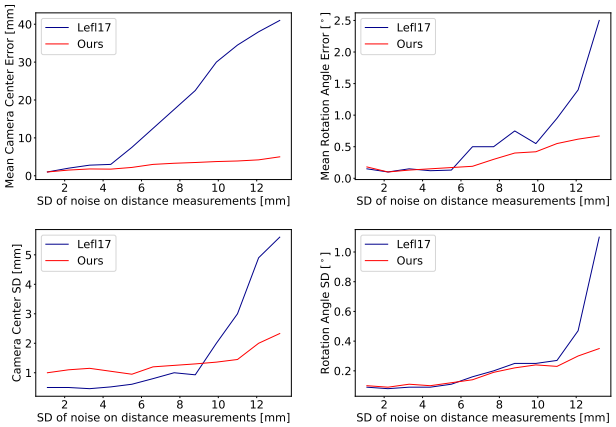


Fig. 9. Comparison of tracking robustness on the *Lego-PAMI-TT Noise Benchmark* [Lefloch et al. 2017]. The mean error (top) and standard deviation for the estimated camera center (bottom-left) and rotation (bottom-right) are evaluated with different levels of (Gaussian) noise. The noisy levels successively increases with standard deviation by integer factors (i.e., 1-13). Note that the camera-distortion based confidence maps [Keller et al. 2013] are employed in [Lefloch et al. 2017].

4.2.2 Photometric term. Following the approach of ElasticFusion [Whelan et al. 2016], color information provided by an RGB-D camera is used to further enhance registration. This complementary information is encoded in a photometric term as

$$E_{color}(\mathbf{T}_i) = \sum_{(\mathbf{u},\bar{\mathbf{u}})\in\Psi} (\bar{\mathbf{C}}_{i-1}(\pi(\mathbf{T}_{i-1}^{-1}\mathbf{T}_i\mathbf{v}_i)) - \mathbf{C}_i(\mathbf{u}))^2, \tag{10}$$

where $\bar{\mathbf{C}}_{i-1}$ and $\mathbf{C}_i$ denote the RGB color value in the predicted map of the previous frame and the color in the current input frame. π is the projection function between 3D objects and the corresponding image frame.

The final objective function to be minimized is

$$E(\mathbf{T}_i) = w_{geom}E_{geom}(\mathbf{T}_i) + E_{color}(\mathbf{T}_i), \tag{11}$$

where w_{geom} is the weight of the geometric term. $w_{geom} = 10$ is suggested in [Whelan et al. 2016] and works well in all our tests. We employ the Gauss-Newton nonlinear least-squares method [Björck 1996] to minimize this energy function, which leads to a reliable alignment between the current input frame and the global model usually after around 20 steps of iteration.

5 DEPTH MAP FUSION

Given a valid camera pose (Section 4), the depth map fusion step integrates the input points and their attributes into a global model as an enriched surfel representation (Section 2).

Let $\mathbf{T}_i \in \mathbb{SE}_3$ denote the pose of i-th input frame, we transform both points and their normals into the $(i-1)$-th frame to conduct the data fusion. Points of the global model are also projected into the $(i-1)$-th frame with their vertex ID stored in the texture map. After that, for each transformed point of the i-th frame, we follow the scheme of [Cao et al. 2018; Keller et al. 2013] to search its *valid* neighbors in a 5×5 window by using the same position / normal compatibility condition. When there are multiple *valid* neighbors, the closest one is chosen to conduct fusion by a confidence-weighted averaging. That is,

$$\begin{aligned} \bar{\mathbf{v}} \leftarrow \frac{\bar{c}\bar{\mathbf{v}} + c\mathbf{v}_{g,i}}{\bar{c}+c}, \bar{\mathbf{n}} \leftarrow \frac{\bar{c}\bar{\mathbf{n}} + c\mathbf{n}_{g,i}}{\bar{c}+c}, \\ \bar{\kappa}_1 \leftarrow \frac{\bar{c}\bar{\kappa}_1 + c\kappa_{1,i}}{\bar{c}+c}, \bar{\kappa}_2 \leftarrow \frac{\bar{c}\bar{\kappa}_2 + c\kappa_{2,i}}{\bar{c}+c}, \\ \bar{r} \leftarrow \bar{c}\bar{r} + cr, \bar{c} \leftarrow \bar{c} + c, \bar{t} \leftarrow t_i, \end{aligned} \tag{12}$$

with $\mathbf{v}_{g,i}$ and $\mathbf{n}_{g,i}$ being the position and normal of an input point in the global model's coordinate. $\bar{c}$, $\bar{\kappa}_1$ and $\bar{\kappa}_2$ are the stored confidence and curvature values of a point on the global model, and c, $\kappa_{1,i}$ and $\kappa_{2,i}$ are the values on an input point evaluated by using the on-the-fly HRBF implicits (Section 3.2 and 3.3).

Points of global model with confidence above a threshold σ_{conf} are considered as *stable* points (e.g., $\sigma_{conf} = 5.0$ is employed by following [Cao et al. 2018; Keller et al. 2013]), and only stable points are used for HRBF surface prediction (Section 3.1). If no corresponding model point is identified, we add the current vertex with its attributes to the global model as an unstable point. Besides, we remove points with confidence values below this threshold for a period of time (i.e., 200 frames) by considering them as noises or outliers.

6 RESULTS AND DISCUSSION

We have implemented our algorithm[2] in the framework of ElasticFusion [Whelan et al. 2016] by C++, CUDA, and OpenGL Shading Language. Moreover, we have incorporated *ORB-SLAM2* [Mur-Artal and Tardós 2017] into our system for the implementation of submap-based hierarchical optimization for large-scale scanning. Our system has been evaluated on both synthetic datasets and raw sequences captured by various depth cameras, including structured light cameras (e.g., *Asus XTion PRO LIVE*, *PrimeSense Carmine* and *Microsoft Kinect v1*) as well as Time-of-flight cameras (e.g., *Microsoft Kinect v2*). We carried out all our experiments on a desktop PC equipped with an Intel Core i7-9700K CPU @3.60GHz with 16GB RAM and a GeForce RTX 2070 GPU with 8GB memory. In this section, we first briefly describe the datasets. Then we present our visual results, followed by the evaluation of our method on different datasets. The output of our system can be either a point cloud or a triangular mesh extracted from the iso-surface maintained by the closed-form HRBF representation using the dual-contouring method [Liu et al. 2016]. Figure 10 shows some small and middle-sized objects rendered in meshes. While in all other figures, we directly render point clouds for the sake of efficiency. All reconstructed 3D models are visualized by Easy3D [Nan 2021], which is an open-source library for 3D modeling, geometry processing, and rendering.

[2]The source code is available at: https://github.com/YabinXuTUD/HRBFFusion3D.

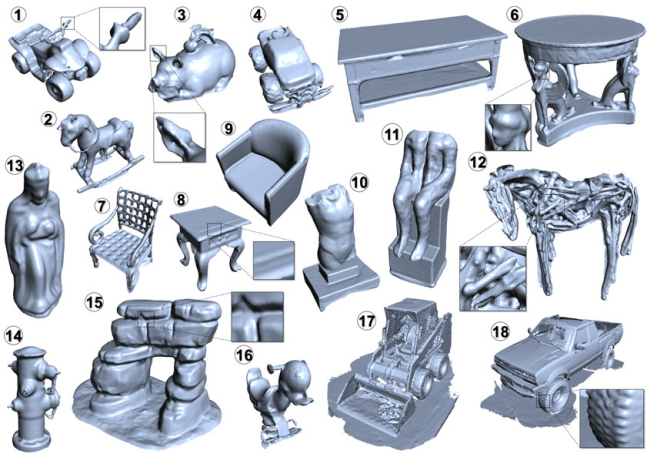


Fig. 10. Reconstructed 3D individual models with different geometric properties and scalability from the *Object Scans* [Choi et al. 2016] dataset. Here the models are rendered by polygonal meshes extracted from the iso-surfaces of HRBF implicits.

6.1 Test datasets

We tested our method on the following datasets.

6.1.1 Object Scans [Choi et al. 2016]. This dataset provides more than 10,000 individual 3D object scans that contain a diversity of objects with different geometric properties and scalability. The scans are captured by unprofessional operators with a PrimeSense Carmine RGB-D camera.

6.1.2 ICL-NUIM benchmark [Handa et al. 2014]. This is a synthetic benchmark dataset with geometric and camera pose ground truth. We selected four scenes of *living rooms* (including synthetic noise) commonly used in the previous work to evaluate the tracking accuracy and reconstruction quality of our results.

6.1.3 TUM benchmark [Sturm et al. 2012]. It is a dataset captured by a *Microsoft Kinect v1* with motion-captured camera poses as ground truth, which is widely used to evaluate the tracking accuracy of a reconstruction method. We select four frequently used sequences (i.e., *fr1/desk*, *fr2/xyz*, *fr3/office*, *fr3/nst*) for the evaluation.

6.1.4 CoRBS benchmark [Wasenmüller et al. 2016]. This is a benchmark dataset of *Microsoft Kinect v2* providing both the motion-captured camera poses and the 3D models acquired by a high-precision commercial scanner as ground-truth. We select the *human* model (Fig. 8) and the *racing car* model (in supplementary video) to demonstrate the performance of our approach.

6.1.5 CuFusion dataset [Zhang and Hu 2017]. This dataset contains both synthetic and real-world sequences for object scanning. Both ground truth trajectories and 3D models are provided on the synthetic examples. We select the synthetic sequence *Armadillo* (that does not have color information) for the evaluation.

6.1.6 ScanNet dataset [Dai et al. 2017]. This dataset is an RGB-D video dataset captured by structure sensors, which consists of 2.5

million views in more than 1,500 scanned sequences. We randomly selected 200 sequences to test the performance of our approach.

6.1.7 Our dataset. We scanned a few objects and large indoor scenes using a *Microsoft Kinect v1* and are shown in Figs. 12, 16, 13, 20, and 21. This is mainly used to evaluate the detail recovery and the scalability of our method. For the evaluation of reconstruction accuracy, we obtain the ground truth models shown in Fig. 13 by a commercial hand-held structure light scanner, Artec Eva, with the precision of $0.1mm$.

6.2 Visual results

6.2.1 Individual objects. We first tested our method on a variety of objects from the *Object Scans* dataset [Choi et al. 2016]. Figure 10 shows the reconstruction results of 18 objects of different sizes and characteristics. Among these objects, (1), (2), (3), (4) are small toys, where the average size is about $0.56m \times 0.30m \times 0.36m$, small geometric features are presented (i.e., the handlebar in (1) with a radius of $0.01m$, the ear in (3) with a thickness of $0.02m$ (as shown in zoom-views of Fig. 10). It is intractable for the methods based on a volumetric representation to reconstruct such geometric details while still adapting to the scale of its background. On the contrary, our HRBF-based on-the-fly surface representation has addressed such limitation since the reconstruction quality only depends on the local kernels and the corresponding support radius (Section 3.1).

Apart from the small-sized toys, we also tested our system on middle-sized objects, including indoor furniture (5) (6) (7) (8) (9), sculpture (10) (11) (12) (13), and outdoor equipment (14) (15) (16). The average size is around $0.89m \times 0.60m \times 1.03m$. The chair (9), the sculpture (10) (11), and the outdoor equipment (15) mainly demonstrate curved surfaces while the tables (5) (6) (8) contain large planar regions. Besides, the chair (7) and the horse models (12) have dense tube-like structures, which poses challenges for RGB-D reconstruction systems. Thanks to the high adaptivity provided by the HRBF on-the-fly surface representation, such fine geometric features (i.e., the decoration on the legs of table (6), the small crease on the desktop in (8), and the concave part in (15) (see the zoom-views in Fig. 10)) are faithfully recovered by our system.

At last, we tested our system with relatively large vehicles (17) and (18), the sizes of which are $4.94m \times 1.97m \times 1.94m$ and $3.33m \times 1.68m \times 1.94m$ respectively. Our system can reconstruct not only global consistent models but also fine geometric details. This can be observed from the crease of the tires on both objects (as shown in the zoom-view in Fig. 10).

6.2.2 Large scenes. Figure 1 presents two large-scale indoor scenes reconstructed by our system. The left shows the reconstructed results of a study room in a university library, while the right shows a study platform in a grand hall of an academic building. Please note that the length of both scenes is above $21m$. Due to the complexity of the scene layout, the camera trajectories are extremely complicated, posing challenges to both camera tracking and reconstruction. The detailed camera trajectories can be found in our supplementary video. Our system managed to capture and reconstruct both scenes with high fidelity.

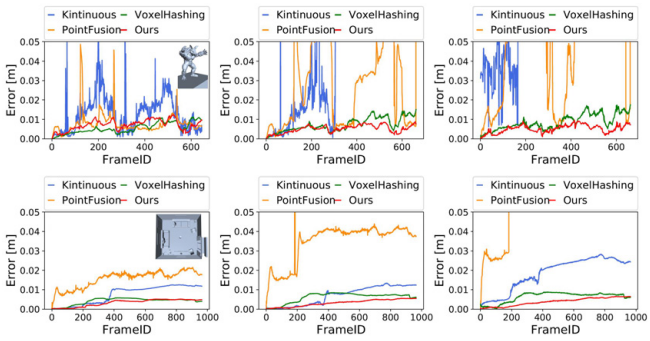


Fig. 11. Comparison of the influence of different representations on tracking robustness with *Kintinuous* [Whelan et al. 2012], *PointFusion* [Keller et al. 2013] and *VoxelHashing* [Niessner et al. 2013] on two noisy scanning sequences – (top row) the *Armadillo* model of the *CuFusion Dataset* [Zhang and Hu 2017] and (bottom row) the scene sequence *lr kt1* of the *ICL-NUIM dataset* [Handa et al. 2014]. From left to right, noises are added into the depth maps in different levels of normal distribution: $\sigma = 3.0$, $\sigma = 6.0$ and $\sigma = 12.0$. Note that we clip tracking error larger than 0.05m and consider it as tracking lost. The insets show the ground truth-geometry of test data.

6.3 Evaluation

In addition to the above visual results, we also conducted a comprehensive analysis of our method in terms of tracking robustness, detail recovery, scalability, reconstruction accuracy, ablation study, parameter discussion, memory consumption, and processing times. Details are given below.

6.3.1 Tracking robustness. The camera tracking of RGB-D reconstruction systems generally tends to drift due to the noise and sparsity in the input frames, which also accumulates noise in the global model. We evaluated the performance of our HRBF-based surface evaluation in tracking robustness below.

Three state-of-the-art reconstruction systems are selected to compare the influence of different representations on tracking robustness, including *Kintinuous* [Whelan et al. 2012], *PointFusion* [Keller et al. 2013], and *VoxelHashing* [Niessner et al. 2013]. *Kintinuous* is an extended version of the original *KinectFusion* [Newcombe et al. 2011] by exploiting a dynamic volume. *VoxelHashing* utilizes a hashing structure to maintain a sparse representation with voxel grids. *PointFusion* uses a surfel representation for camera tracking. The experiment is conducted on two synthetic sequences with ground truth camera poses: the *Armadillo* of the *CuFusion Dataset* [Zhang and Hu 2017] and the *lr kt1* of the *ICL-NUIM dataset* [Handa et al. 2014]. Noises are added in different levels of normal distribution (i.e., $\sigma = 3.0$, $\sigma = 6.0$ and $\sigma = 12.0$) to test the robustness of different systems. We evaluated the camera pose error for all frames and the results are shown in Fig. 11. It can be found that the point-based representation is more sensitive to noise while our HRBF-based method demonstrates consistently low errors in camera tracking.

We further evaluated our system in terms of accuracy in trajectory estimation on the *TUM benchmark* [Sturm et al. 2012] (*Microsoft Kinect v1*) where ground truth trajectories are provided by a highly accurate calibrated motion-capture system. We chose a set of widely used sequences (i.e, *fr1/desk*, *fr2/xyz*, *fr3/office*, *fr3/nst*) and compared

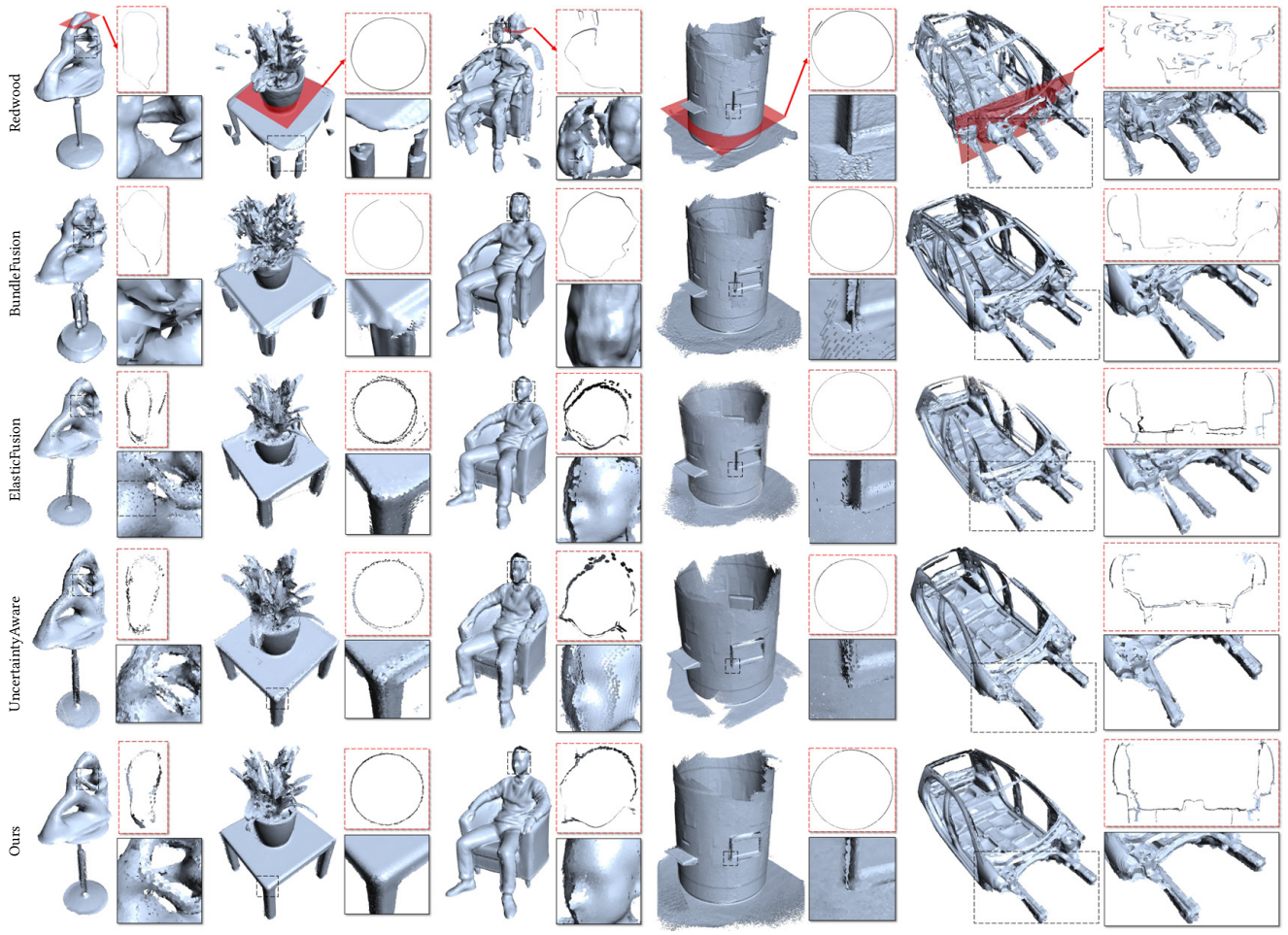


Fig. 12. Comparison of reconstruction results generated by *Redwood* [Choi et al. 2016], *BundleFusion* [Dai et al. 2017], *ElasticFusion* [Whelan et al. 2016], *UncertaintyAware* [Cao et al. 2018], and our method on five objects with different geometric shapes and details. Models from left to right are: *Fertility*, *Plant*, *Human*, *Pillar* and *Car Frame*. Note that the *Redwood* and the *BundleFusion* methods generates mesh surfaces from volume representation as results (displayed in the first two rows) while the results of other three methods as point clouds are rendered by surfel splatting.

our methods with state-of-the-art online reconstruction systems, including *DVO-SLAM* [Kerl et al. 2013], *RGBD SLAM* [Endres et al. 2012], *MRSMap* [Stückler and Behnke 2014], *Kintinuous* [Whelan et al. 2012], *ElasticFusion* [Whelan et al. 2016], *BundleFusion* [Dai et al. 2017], and *UncertaintyAware* [Cao et al. 2018]. To make a complete comparison, the offline reconstruction system, *Redwood* [Choi et al. 2015], is also included. We recorded the absolute trajectory error (ATE) of root-mean-square error (RMSE) for camera tracking accuracy. The results are summarized in Table 1. We can see that our method consistently outperformed (or demonstrated comparable) results to the most promising methods in the comparison. To analyze camera tracking drift, we separately evaluated our method with and without global optimization (i.e., similar to *UncertaintyAware* [Cao et al. 2018] that both local and global bundle adjustment (BA) are applied in global optimization). Most existing systems have applied different global optimization techniques to alleviate the accumulated errors in camera pose estimation.

- *DVO SLAM*, *RGBD SLAM*, and *MRSMap* first apply a pose graph optimization to achieve a global consistent trajectory and then the global model is constructed by integrating all depth maps in a volumetric representation (i.e., *DVO SLAM* and *RGBD SLAM*) or merging key surfel views (i.e., *MRSMap*).
- *Kintinuous* and *ElasticFusion* achieve a globally consistent model in a map-centric manner by deforming the global model according to global or local constraints.
- *Redwood*, *BundleFusion*, and *UncertaintyAware* divide the global model into submaps and obtain a globally consistent model by optimizing between submaps.

Our system outperforms most of these systems. There is one exception that *BundleFusion* achieved the best result on *fr3/nst*. The main reason lies in its combined sparse visual features, dense

Table 1. ATE RMSE on the TUM benchmark (unit: m)

	fr1/desk	*fr2/xyz*	*fr3/office*	*fr3/nst*
DVO SLAM	0.021	0.018	0.035	0.018
RGBD SLAM	0.023	0.008	0.032	0.017
MRSMap	0.043	0.020	0.042	2.018
Kintinuous	0.037	0.029	0.030	0.031
ElasticFusion	0.020	0.011	0.017	0.016
BundleFusion	0.016	0.011	0.022	**0.012**
Redwood	0.027	0.091	0.030	1.929
UncertaintyAware	0.015	0.006	0.009	0.014
Ours	**0.014**	**0.005**	**0.007**	0.016
Comparison of only applying Local BA				
UncertaintyAware	0.015	0.006	0.037	0.014
Ours	0.014	0.005	0.015	0.016
Comparison of only applying Global BA				
UncertaintyAware	0.033	0.009	0.025	0.093
Ours	0.018	0.007	0.014	0.030

photometric and geometric objective, which enables it to obtain a more tight alignment on textured scenes. Global optimization techniques such as local or global BA can help significantly reduce the tracking errors in practice. It is interesting to compare the errors after removing either local or global BA (see the last two parts of Table 1). We can find that our results are more accurate than those of *UncertaintyAware* in most cases.

6.3.2 Detail recovery. With the help of on-the-fly HRBF surface representation, our method is able to recover finer geometric details. To demonstrate this capability, we compared our method with the state-of-the-art reconstruction systems including *Redwood* [Choi et al. 2015], *ElasticFusion* [Whelan et al. 2016], *BundleFusion* [Dai et al. 2017], and *UncertaintyAware* [Cao et al. 2018] on a variety of 3D objects (see Fig.12). Since our scanning aims at achieving a complete model, a global loop is required to exist for every model.

The *Redwood* system [Choi et al. 2015] cannot generate good results due to the registration error. It completely failed on the human example (see the human face and the right leg in the third column). The same issue of camera tracking drift also occurs in the *ElasticFusion* and the *BundleFusion* systems. In short, all these three systems are unable to produce a global consistent 3D model. The *UncertaintyAware* approach can obtain global consistent models by successfully detecting the close loop in all examples. However, artifacts are still generated by the *UncertaintyAware* approach due to the accumulated error – see the human face in the third column. As has been expected, our HRBF-based method is more robust in recovering geometric details.

6.3.3 Reconstruction accuracy. To evaluate the reconstruction accuracy, we compared the results of *Redwood* [Choi et al. 2015], *ElasticFusion* [Whelan et al. 2016], *BundleFusion* [Dai et al. 2017], *UncertaintyAware* [Cao et al. 2018], and ours to the 3D models acquired by a commercial hand-held structure light scanner, Artec Eva, with the precision of 0.1*mm*. The model obtained from this structure light scanner is referred to as ground truth. To evaluate the relevant scenery, we manually removed the background of the obtained model from each method. Each model is aligned to the ground truth mesh and the distance error is computed and visualized

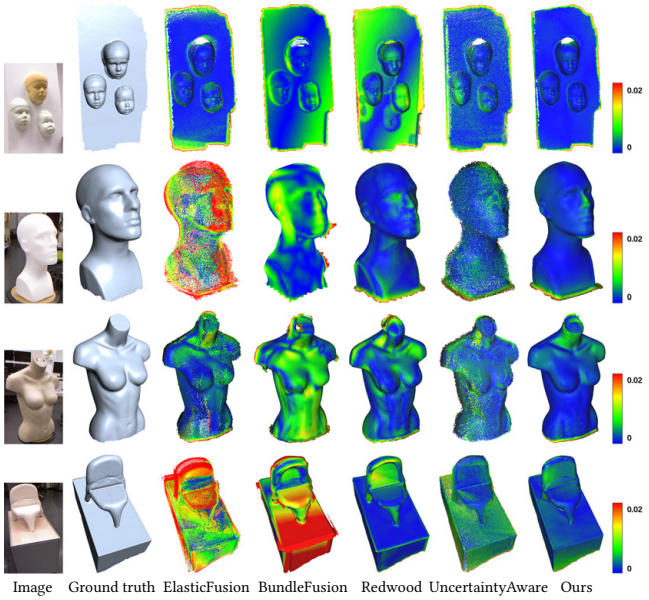


Fig. 13. Comparison of the reconstruction accuracy with *ElasticFusion* [Whelan et al. 2016] (third column), *BundleFusion* [Dai et al. 2017] (fourth column), *Redwood* [Choi et al. 2015] (fifth column), and *UncertaintyAware* (sixth column). The ground-truth models (second column) were obtained by a high-precision structure light 3D scanner. The color map presents the distance error on the reconstructed models. Models from top to bottom are: *Faces*, *Head*, *Upper Body*, *Small Chair*.

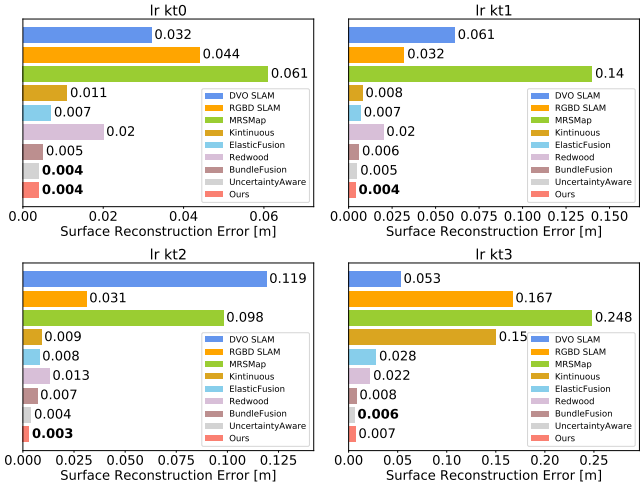


Fig. 14. Surface reconstruction error in terms of average point to surface distances on the *ICL-NUIM benchmark* (unit: meter). The best performance is highlighted in bold fonts.

as a color map (see Fig. 13). As can be observed, all these methods were able to produce consistent 3D models and our results have the smallest errors while preserving more geometric details than the other methods. The errors were mainly sourced to camera tracking, which is prone to noises on the input RGB-D images. By using

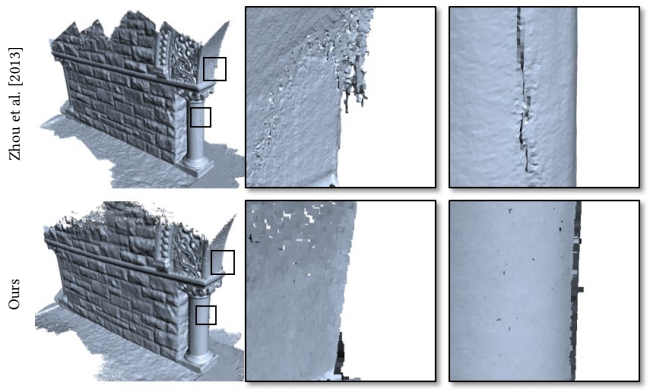


Fig. 15. Comparison of the reconstruction quality between the offline optimization based method of Zhou et al. [2013] (top) and our approach (bottom) on the *stonewall* example from their 3D Scene Dataset. This example consists of 2, 700 frames.

the on-the-fly HRBF surface estimation together with the weighted registration strategy (i.e., curvature, confidence, and photometric), our system is more robust in camera tracking, therefore, yielding the highest precision among all these systems.

We also evaluate the surface reconstruction accuracy in terms of average point-to-surface distances on the *living room kr0-kr3* models from the *ICL-NUIM benchmark* [Handa et al. 2014]. Our method is compared with a variety of existing approaches and the results are summarized in Fig. 14. It is easy to find that our method can achieve better (or comparable) results in terms of reconstruction accuracy. Again this is benefited from the robust HRBF on-the-fly surface estimation presented in this paper.

6.3.4 Scalability. With the robustness in camera tracking and surface prediction, our method can reconstruct large scenes. In addition to the two scenes already shown in Fig. 1, we tested our approach on the *stonewall* models from the 3D Scene dataset (see Fig. 15). Comparing their method with the offline global optimizer [Zhou and Koltun 2013], we can observe a significant reduction of camera tracking drift on our result on this example with 2, 700 frames.

As shown in Figs. 16 and 17, we captured a sequence of 6, 114 RGB-D images in a conference room by a *Microsoft Kinect v1* camera with a complex camera trajectory. The trajectory contains many local loops. When comparing with other state-of-the-art reconstruction systems including *Redwood* [Zhou and Koltun 2015], *ElasticFusion* [Whelan et al. 2016], *BundleFusion* [Dai et al. 2017]), and UncertaintyAware [Cao et al. 2018], all the other four methods suffer from camera tracking drift (especially in the regions with local loops on the trajectory) and perform poorly in recovering surface details – see the 'double-layers' of chairs (fourth column) and tables (fifth column) shown in the zoom-views. Our robust surface estimation by using on-the-fly HRBF implicits can effectively reduce the error in camera tracking drift thus can generate more consistent 3D reconstruction.

We also conducted experiments on the ScanNet dataset [Dai et al. 2017]. Among its 1,500 scan sequences, we randomly selected 200 sequences to reconstruct 3D scenes and compared our results with those from BundleFusion [Dai et al. 2017]. It is found that similar results are generated by both methods on most of the sequences especially on those for simple scenes. Better reconstruction results can be found on 5 sequences with complex trajectories. For example, in the scene shown in Fig. 18, the structural distortion is significantly reduced by our method. Similar improvement can also be found on the other four scenes as shown in Fig. 19.

Moreover, we captured a sequence of 14, 163 RGB-D frames with a quite long trajectory on an urban street using a *Microsoft Kinect v1*. We compare the reconstruction results with *BundleFusion* [Dai et al. 2017] and *UncertaintyAware* [Cao et al. 2018] in Fig. 20. We can observe that the result of *BundleFusion* [Dai et al. 2017] breaks (see the zoom-view on the top left) and fails to generate a globally consistent 3D model. *UncertaintyAware* can obtain a more consistent result but still suffers from camera tracking drift, which leads to artifacts in the reconstruction (see the zoom-view on the right of the second row). In contrast, our system can produce a globally consistent 3D model. It is also worthy to note that surfel-based representation has the advantage to preserve geometric details. This can be observed from the number plate '1' in the right zoom-views, where the result obtained from the volumetric representation of *BundleFusion* is not as clear as *UncertaintyAware* and ours.

6.3.5 Ablation Study. We further conducted an ablation study to evaluate the effectiveness of each single algorithm component of our system by replacing it with another option used in others' work, where the study is taken on a sequence of 4, 080 RGB-D images captured in a meeting room (Fig. 21). For quantitative analysis, we also plot the mean distance errors of all validated vertex pairs for each frame pair to indicate the quality of the registration as shown in Fig. 21(f). Due to the high sensitivity to noise, the surfel-based representation led to a dramatic increase in mean distance error (see Fig. 21(f)) and unsatisfactory reconstruction (see the close-up view shown in Fig. 21(a)). In the second test, we replace our HRBF-based curvature estimation with the method presented in [Lefloch et al. 2017]. Although the global consistent 3D model can be obtained – thanks to global techniques of local and global BA, noises induced by the black surfaces (i.e., chairs) can lead to artifacts in intra-submap level as shown in Fig. 21(b). In Fig. 21(c), we utilize the camera-distortion based evaluation method [Keller et al. 2013] for confidence map. As a result, the accumulated noise in the global model leads to unstable registration and imperfect reconstruction. The last two tests are conducted to evaluate the importance of local BA (Fig. 21(d)) and global BA (Fig. 21(e)) in our pipeline of reconstruction, where local BA helps to recover the artifacts between submaps and global BA helps to generate globally consistent models.

6.3.6 Parameter discussion. As a key parameter of our system, the support size influences the accuracy of registration. We select different sizes of window patches (Section 3.1) as 5×5, 7×7 (default), and 9×9 for the experiment. The evaluation is conducted on three different datasets, i.e., *TUM benchmark*, *ICL-NUIM benchmark*, and *CoRBS benchmark*, and the results are presented in Fig. 22. A larger support size leads to a smoother surface while a smaller support

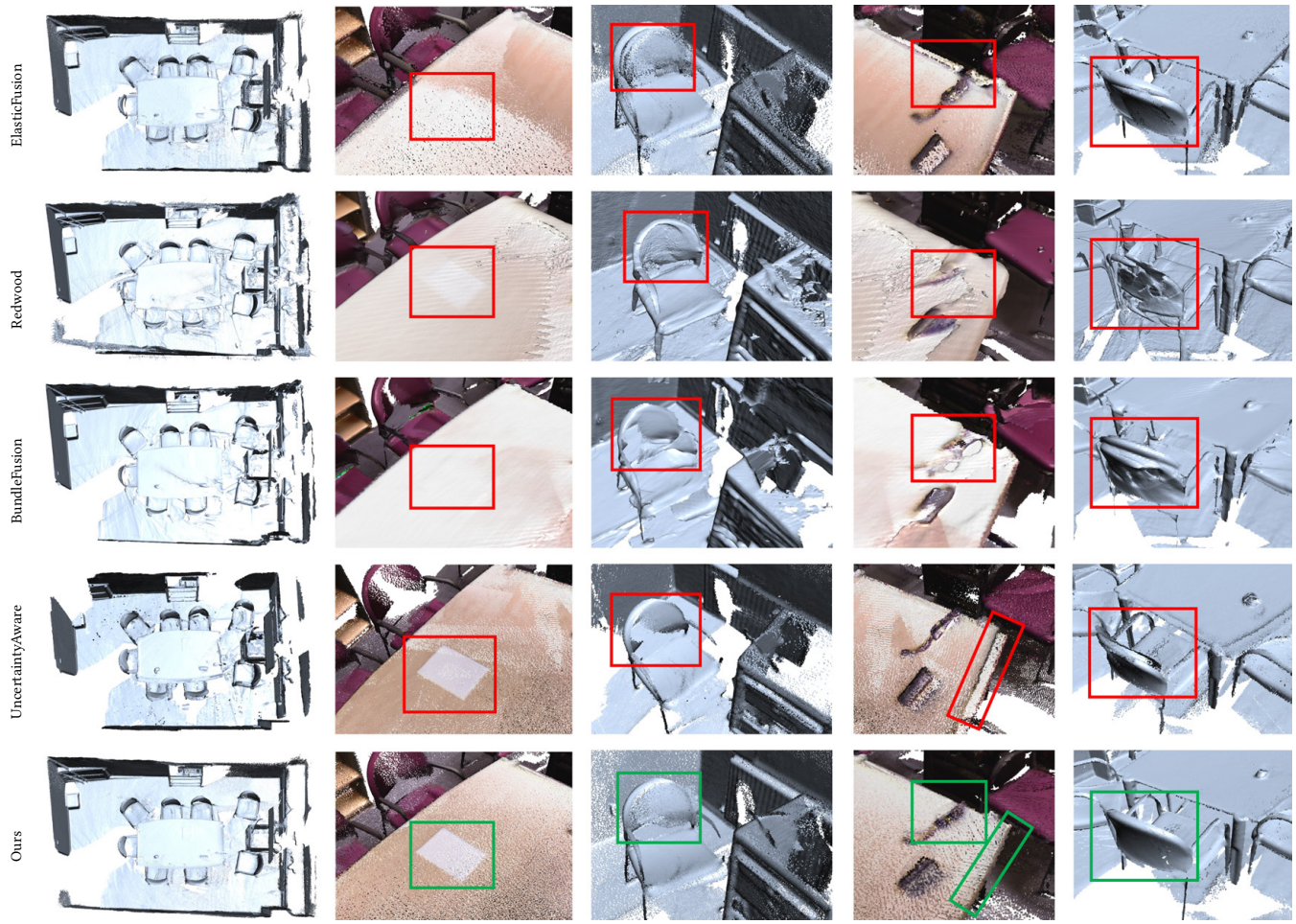


Fig. 16. Comparison with state-of-the-art RGB-D reconstruction systems, i.e., *ElasticFusion* [Whelan et al. 2016], *Redwood* [Choi et al. 2015], *BundleFusion* [Dai et al. 2017], and UncertaintyAware [Cao et al. 2018] on a sequence of 6, 114 RGB-D images captured in a conference room by a complex camera trajectory that consists of many local loops (see Fig. 17).

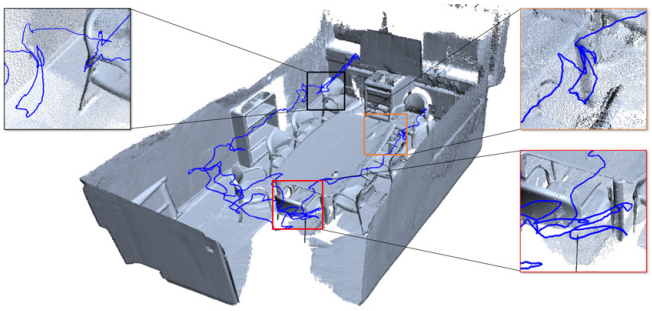


Fig. 17. The complex camera trajectory for the conference room example shown in Fig. 16 with 6, 114 frames.

size can preserve more geometric details. Correspondingly, a patch size of 5×5 is suitable for the reconstruction of clean data but not robust enough to handle the noises induced by depth cameras. On the other hand, a patch size of 9×9 always leads to over-smoothing results. A patch size of 7×7 can achieve the best performance in our tests as shown in Fig. 22.

6.3.7 Performance. To study the memory consumption of our approach, we recorded the GPU memory consumption for storing and managing the global model over 13 sequences of RGB-D images that are captured. Comparison with the *BundleFusion* system [Dai et al. 2017] is conducted to demonstrate the memory efficiency of our HRBF-based method – see Table 2. Note that *BundleFusion* shares the same representation with *VoxelHashing* [Niessner et al. 2013], which exploits sparsity by applying a hash-based structure to the volumetric representation. The memory consumption of *BundleFusion* by using two different voxel sizes is reported. When 4mm is used for the voxel size – being able to capture more geometric details, the *BundleFusion* system failed to add depth maps during reconstruction due to the large memory requirement. Our system has a significantly smaller memory footprint compared to the

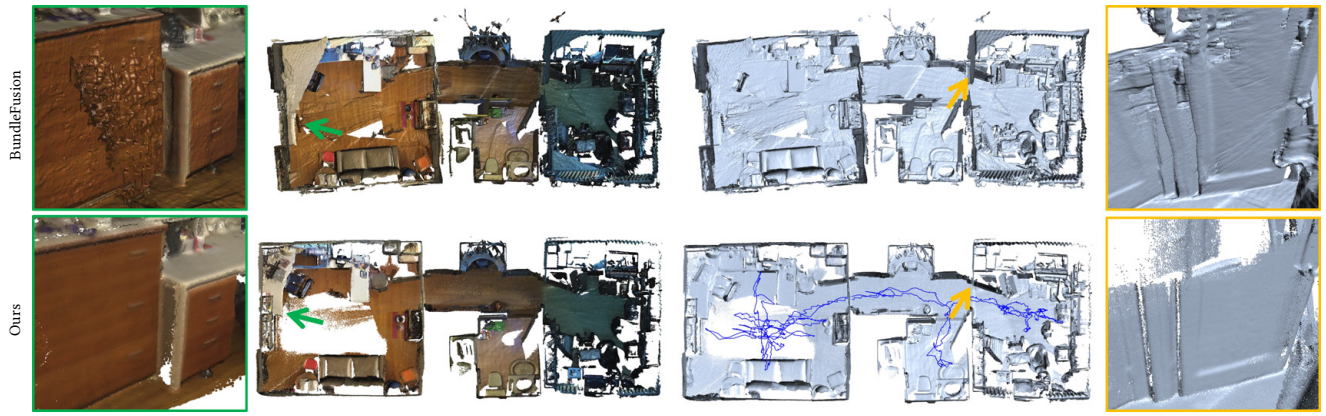
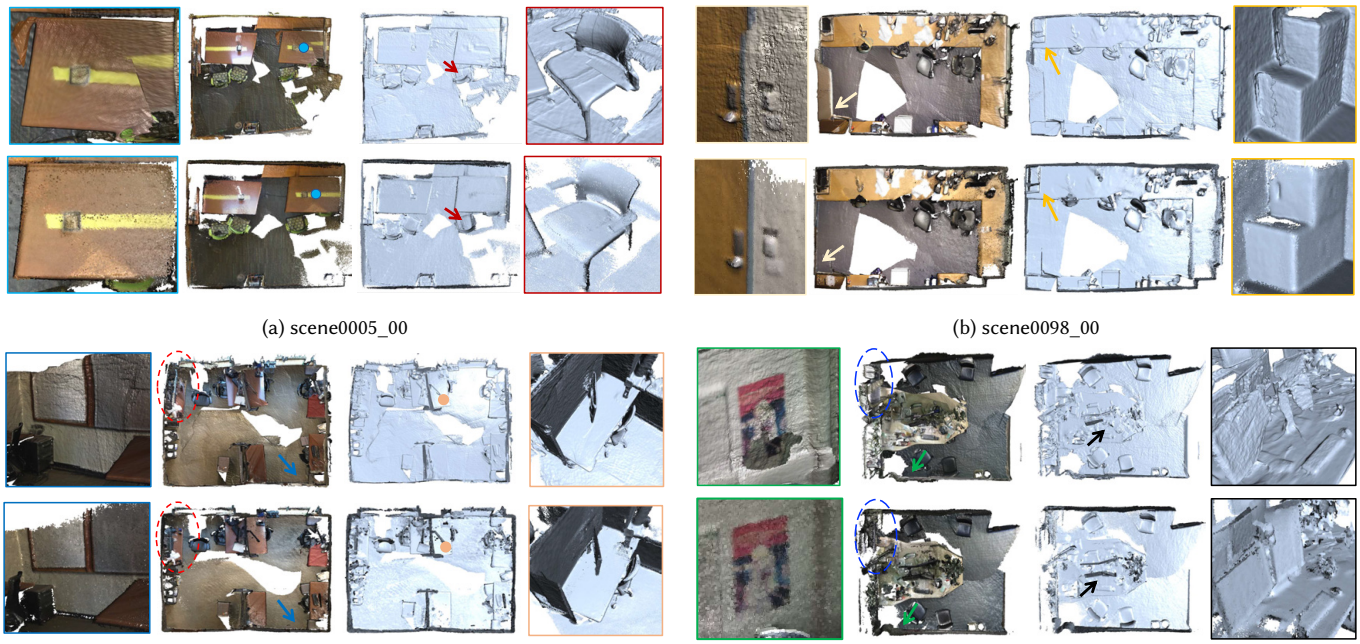


Fig. 18. Comparison of our reconstruction result (bottom row) with BundleFusion [Dai et al. 2017] (top row) on "scene0054_00", a sequence of 6,629 RGB-D frames captured by a structure sensor, from ScanNet [Dai et al. 2017]. Closer inspections (green and yellow boxes) are presented to show reconstruction details of each method. The camera trajectory is visualized in blue color. Our approach maintains only global model points with confidence values larger than a threshold (see Section 5) similar to [Cao et al. 2018; Keller et al. 2013; Whelan et al. 2016], which causes some missing data on the floor.



(a) scene0005_00 (b) scene0098_00

(c) scene0142_00 (d) scene0017_00

Fig. 19. More comparison between the reconstruction results of BundleFusion [Dai et al. 2017] (top) and ours (bottom) on the other four sequences from the ScanNet dataset [Dai et al. 2017], which contain 1159, 1285, 2434, and 1490 RGB-D frames, respectively. Structural distortions are marked in colored circle, and closer zoom-views are also presented to show the details of 3D reconstruction.

volumetric representation based approaches and therefore is more suitable for reconstructing large scenes.

We report the computing time used by each component of our system in Fig. 23 for all RGB-D frames throughout the sequence of *Meeting Room* (Fig. 16). The efficiency of different components of the computational pipeline has been analyzed. In general, our system can achieve an average processing time of 42ms per frame, which indicates a near real-time performance (i.e., approximately 24Hz). Among all components of our system, the HRBF-based prediction takes over half of the processing time (25ms). For comparisons, we also plot the processing times of *BundleFusion* [Dai et al. 2017] (left) and *UncertaintyAware* [Cao et al. 2018] (center) in Fig. 23.

Fig. 20. Comparison of our reconstruction result (bottom row) with state-of-the-art RGB-D reconstruction systems, i.e., *BundleFusion* [Dai et al. 2017] (top row) and *UncertaintyAware* [Cao et al. 2018] (center row) on a sequence of 14, 163 RGB-D frames captured on an urban street with a long trajectory. Closer inspections (black boxes) are presented to show the reconstructed details of each method.

Table 2. Statistic of memory consumption for reconstruction (unit: MB)

Model Name	Fig.	#Frames	BundleFusion (GPU)		Ours (GPU)
			Voxel Size		
			10mm	4mm	
Fertility	12	1,301	153.0	740.9	15.0
Plant	12	1,703	162.9	830.5	25.3
Human	12	2,751	137.4	1309.1	44.9
Pillar	12	1,987	198.2	1536.6	59.8
Car Frame	12	3,694	126.8	1462.5	95.1
Faces	13	579	59.0	443.0	21.1
Head	13	1,663	167.6	1806.4	15.6
Upper Body	13	1,308	179.7	1951.6	22.3
Small Chair	13	1,693	107.5	1115.2	34.0
Conference Room	16	6,114	164.7	1240.3	98.1
Urban Street	20	14,163	2205.3	-	552.5
Library	1	16,128	1924.1	-	571.3
Study Platform	1	10,930	1835.0	-	603.1

7 CONCLUSION AND FUTURE WORK

We have presented the HRBF-Fusion as a new method using on-the-fly HRBF implicits for 3D reconstruction from RGB-D images. Our system is not only able to reconstruct objects with high fidelity but also scalable to large scenes after incorporating submap-based local and global optimization strategies. The robustness of our HRBF-Fusion is mainly due to the robust curvature estimation based on the HRBF implicits, which can significantly reduce the drift in camera tracking. Moreover, our reconstruction-indicated surface evaluation method exploits the uncertainty of the measurement in the input depth maps and further improves the accuracy in both the camera tracking step and the finally reconstructed models. The surfel representation using on-the-fly HRBF implicits has a low memory footprint and is suitable for reconstructing large scenes.

The proposed system can reconstruct long-range scanning with submap level local and global optimization. However, camera tracking failure may still happen between intra-submaps for featureless regions (e.g., white planar walls). This is a common problem in all existing RGB-D reconstruction systems. A proactive reconstruction method by using robotic systems is planned to be investigated in our future work. Furthermore, it is also interesting to incorporate geometric primitives or structural regularities (i.e., parallelism or orthogonality) to improve the robustness of the hierarchical optimization for long-range scanning.

ACKNOWLEDGMENTS

Yabin Xu was a visiting PhD student supervised by L. Nan and C.C.L. Wang at TU Delft, and he was also partially supported by the China Scholarship Council and the Faculty of Industrial Design Engineering, TU Delft.

REFERENCES

P. J. Besl and N. D. McKay. 1992. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (Feb 1992), 239–256. https://doi.org/10.1109/34.121791

Å. Björck. 1996. *Numerical Methods for Least Squares Problems.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. https://doi.org/10.1137/1.9781611971484

G. Blais and M. D. Levine. 1995. Registering Multiview Range Data to Create 3D Computer Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17, 8 (Aug 1995), 820–824. https://doi.org/10.1109/34.400574

A. Bozic, P. Palafox, J. Thies, A. Dai, and M. Nießner. 2021. Transformerfusion: Monocular rgb scene reconstruction using transformers. *Advances in Neural Information Processing Systems* 34 (2021).

Y. Cao, L. Kobbelt, and S. Hu. 2018. Real-time High-accuracy Three-Dimensional Reconstruction with Consumer RGB-D Cameras. *ACM Trans. Graph.* 37, 5, Article

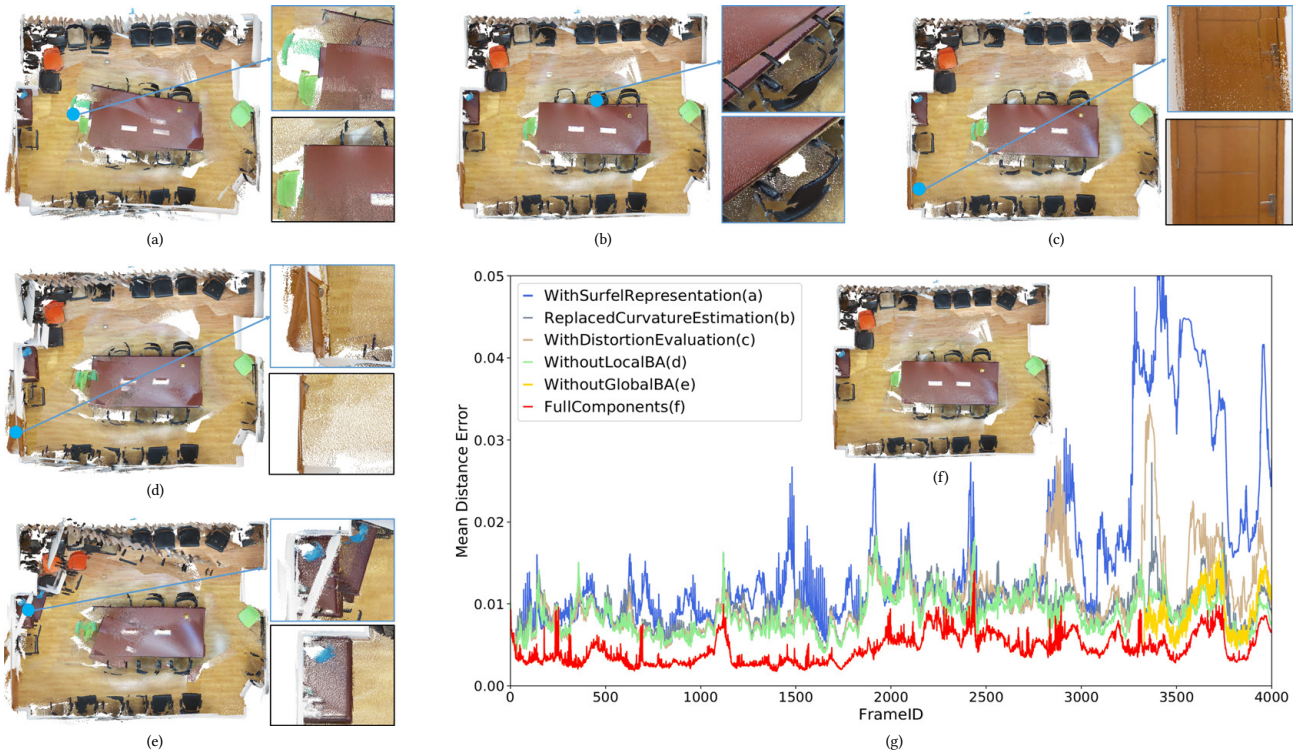


Fig. 21. Evaluation of our system with different options for each single component on a sequence of 4,080 RGB-D images captured in another meeting room (different from Fig. 17). (a) Replacing HRBF implicits by the surfel-based representation. (b) Changing HRBF-based curvature to the curvature estimation method presented in [Lefloch et al. 2017]. (c) Using the camera-distortion based confidence evaluation method in [Keller et al. 2013]. (d) Removing local BA. (e) Removing global BA. (f) Our reconstruction result with full components. (g) Mean distance error of all validated vertex pairs (see in Section 4.1) for each frame pair. The close-up views in (a)-(e) show the artifacts obtained by changing one component (top) and our corresponding reconstruction (bottom) using all algorithm components.

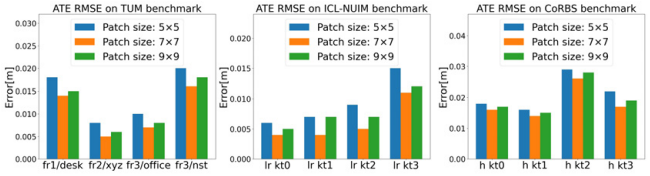


Fig. 22. The evaluation of using different support sizes on the accuracy of camera tracking on different datasets, i.e., *TUM benchmark*, *ICL-NUIM benchmark*, and *CoRBS benchmark*. We select different sizes of window patches (as described in Section 3.1): 5×5, 7×7, and 9×9 for the evaluation.

171 (Sept. 2018), 16 pages. https://doi.org/10.1145/3182157

J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. 2001. Reconstruction and Representation of 3D Objects with Radial Basis Functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 67–76. https://doi.org/10.1145/383259.383266

J. Chen, D. Bautembach, and S. Izadi. 2013. Scalable Real-time Volumetric Surface Reconstruction. *ACM Trans. Graph.* 32, 4, Article 113 (July 2013), 16 pages. https://doi.org/10.1145/2461912.2461940

S. Choi, Q. Zhou, and V. Koltun. 2015. Robust Reconstruction of Indoor Scenes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5556–5565. https://doi.org/10.1109/CVPR.2015.7299195

S. Choi, Q. Zhou, S. Miller, and V. Koltun. 2016. A Large Dataset of Object Scans. arXiv:1602.02481 [cs.CV]

B. Curless and M. Levoy. 1996. A Volumetric Method for Building Complex Models from Range Images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. Association for Computing Machinery, New York, NY, USA, 303–312. https://doi.org/10.1145/237170.237269

A. Dai, A. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. 2017. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2432–2443. https://doi.org/10.1109/CVPR.2017.261

A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. 2017. BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration. *ACM Trans. Graph.* 36, 3, Article 24 (May 2017), 18 pages. https://doi.org/10.1145/3054739

M. Danelljan, G. Meneghetti, F. S. Khan, and M. Felsberg. 2016. A Probabilistic Framework for Color-Based Point Set Registration. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1818–1826. https://doi.org/10.1109/CVPR.2016.201

W. Dong, Q. Wang, X. Wang, and H. Zha. 2018. PSDF Fusion: Probabilistic Signed Distance Function for On-the-fly 3D Data Fusion and Scene Reconstruction. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 714–730.

F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. 2012. An Evaluation of the RGB-D SLAM System. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*. 1691–1696. https://doi.org/10.1109/ICRA.2012.6225199

J. Engel, J. Sturm, and D. Cremers. 2013. Semi-dense Visual Odometry for a Monocular Camera. In *2013 IEEE International Conference on Computer Vision (ICCV)*. 1449–1456. https://doi.org/10.1109/ICCV.2013.183

C. Forster, M. Pizzoli, and D. Scaramuzza. 2014. SVO: Fast Semi-direct Monocular Visual Odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 15–22. https://doi.org/10.1109/ICRA.2014.6906584

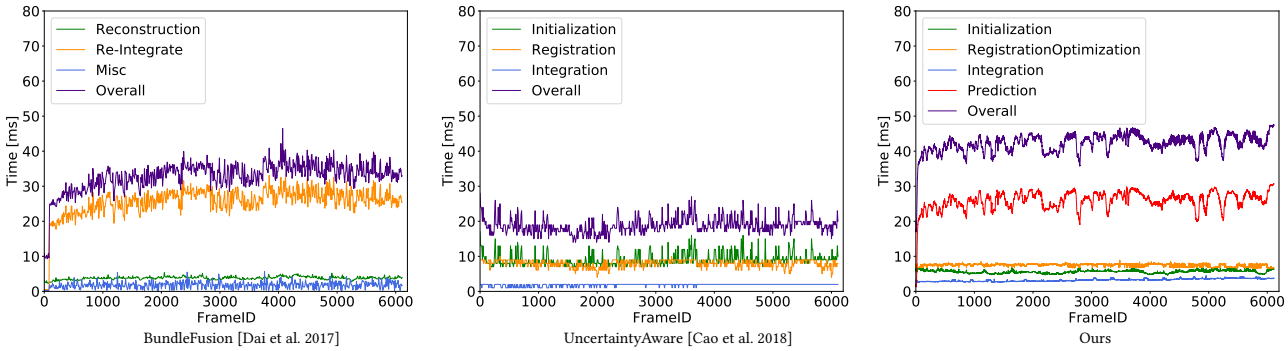


Fig. 23. Comparison of the computing time of our system (right) with *BundleFusion* [Dai et al. 2017] (left) and *UncertaintyAware* [Cao et al. 2018] (center) on each frame of the *Meeting Room* sequence (Fig. 16). The processing time for each main component of the three systems is also plotted. Note that, two GPUs are used for *BundleFusion* as suggested in [Dai et al. 2017] and the time is reported here according to the main GPU. Differently, only one GPU is employed for *UncertaintyAware* and our pipeline.

D. Gallup, M. Pollefeys, and J. Frahm. 2010. 3D Reconstruction Using an N-Layer Heightmap. In *Proceedings of the 32nd DAGM Conference on Pattern Recognition* (Darmstadt, Germany). Springer-Verlag, Berlin, Heidelberg, 1–10.

G. Godin, M. Rioux, and R. Baribeau. 1994. Three-dimensional Registration Using Range and Intensity Information. *Proceedings of SPIE - The International Society for Optical Engineering* 2350 (01 1994), 279–290.

J. Goldfeather and V. Interrante. 2004. A Novel Cubic-Order Algorithm for Approximating Principal Direction Vectors. *ACM Trans. Graph.* 23, 1 (Jan. 2004), 45–63. https://doi.org/10.1145/966131.966134

A. Handa, T. Whelan, J. McDonald, and A. J. Davison. 2014. A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 1524–1531. https://doi.org/10.1109/ICRA.2014.6907054

P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. 2012. RGB-D Mapping: Using Kinect-Style Depth Cameras for Dense 3D Modeling of Indoor Environments. *International Journal of Robotic Research* 31 (04 2012), 647–663. https://doi.org/10.1177/0278364911434148

J. Huang, S. Huang, H. Song, and S. Hu. 2021b. DI-Fusion: Online Implicit 3D Reconstruction with Deep Priors. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*. 8928–8937. https://doi.org/10.1109/CVPR46437.2021.00882

S. Huang, H. Chen, J. Huang, H. Fu, and S. Hu. 2021a. Real-Time Globally Consistent 3D Reconstruction with Semantic Priors. *IEEE Transactions on Visualization and Computer Graphics* (2021), 1–1. https://doi.org/10.1109/TVCG.2021.3137912

B. Jian and B. C. Vemuri. 2011. Robust Point Set Registration Using Gaussian Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 8 (2011), 1633–1645. https://doi.org/10.1109/TPAMI.2010.223

M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb. 2013. Real-Time 3D Reconstruction in Dynamic Scenes Using Point-Based Fusion. In *Proceedings of the 2013 International Conference on 3D Vision (3DV '13)*. IEEE Computer Society, Washington, DC, USA, 1–8. https://doi.org/10.1109/3DV.2013.9

C. Kerl, J. Sturm, and D. Cremers. 2013. Dense Visual SLAM for RGB-D Cameras. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2100–2106. https://doi.org/10.1109/IROS.2013.6696650

G. Klein and D. Murray. 2007. Parallel Tracking and Mapping for Small AR Workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. 225–234. https://doi.org/10.1109/ISMAR.2007.4538852

D. Lefloch, M. Kluge, H. Sarbolandi, T. Weyrich, and A. Kolb. 2017. Comprehensive Use of Curvature for Robust and Accurate Online Surface Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 12 (2017), 2349–2365. https://doi.org/10.1109/TPAMI.2017.2648803

H. Li, E. Vouga, A. Gudym, L. Luo, J. Barron, and G. Gusev. 2013. 3D Self-Portraits. *ACM Trans. Graph.* 32, 6, Article 187 (Nov. 2013), 9 pages. https://doi.org/10.1145/2508363.2508407

L. Liu, Kyaw Z. Gu, J.and Lin, T. Chua, and C. Theobalt. 2020. Neural Sparse Voxel Fields. *NeurIPS* (2020).

S. Liu, H. Guo, H. Pan, P. Wang, X. Tong, and Y. Liu. 2021. Deep Implicit Moving Least-Squares Functions for 3D Reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1788–1797.

S. Liu, C. Wang, G. Brunnett, and J. Wang. 2016. A Closed-Form Formulation of HRBF-Based Surface Reconstruction by Approximate Solution. *Comput. Aided Des.* 78 (Sept. 2016), 147–157. https://doi.org/10.1016/j.cad.2016.05.001

I. Macêdo, J. P. Gois, and L. Velho. 2011. Hermite Radial Basis Functions Implicits. *Computer Graphics Forum* 30, 1 (2011), 27–42. https://doi.org/10.1111/j.1467-8659.2010.01785.x

S. Meerits, D. Thomas, V. Nozick, and H. Saito. 2018. FusionMLS: Highly dynamic 3D reconstruction with consumergrade RGB-D cameras. *Computational Visual Media* 4, 4 (Dec. 2018), 287–303. https://doi.org/10.1007/s41095-018-0121-0

M. Meilland and A. I. Comport. 2013. On Unifying Key-frame and Voxel-based Dense Visual SLAM at Large Scales. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 3677–3683. https://doi.org/10.1109/IROS.2013.6696881

R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. 2015. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics* 31, 5 (2015), 1147–1163. https://doi.org/10.1109/TRO.2015.2463671

R. Mur-Artal and J. D. Tardós. 2017. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics* 33, 5 (Oct 2017), 1255–1262. https://doi.org/10.1109/TRO.2017.2705103

L. Nan. 2021. Easy3D: a lightweight, easy-to-use, and efficient C++ library for processing and rendering 3D data. *Journal of Open Source Software* 6, 64 (2021), 3255. https://doi.org/10.21105/joss.03255

R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. 2011. KinectFusion: Real-time Dense Surface Mapping and Tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. 127–136. https://doi.org/10.1109/ISMAR.2011.6092378

M. Niessner, M. Zollhofer, S. Izadi, and M. Stamminger. 2013. Real-time 3D Reconstruction at Scale Using Voxel Hashing. *ACM Trans. Graph.* 32, 6, Article 169 (Nov. 2013), 11 pages. https://doi.org/10.1145/2508363.2508374

N. Patrikalakis. 2002. *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer-Verlag, Berlin, Heidelberg.

M. Reynolds, J. Doboš, L. Peel, T. Weyrich, and G. J. Brostow. 2011. Capturing Time-of-Flight Data with Confidence. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 945–952. https://doi.org/10.1109/CVPR.2011.5995550

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. 2011. ORB: An Efficient Alternative to SIFT or SURF. In *2011 International Conference on Computer Vision (ICCV)*. 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544

S. Rusinkiewicz and M. Levoy. 2001. Efficient Variants of the ICP Algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. 145–152. https://doi.org/10.1109/IM.2001.924423

H. Sarbolandi, D. Lefloch, and A. Kolb. 2015. Kinect Range Sensing: Structured-light Versus Time-of-Flight Kinect. *Computer Vision and Image Understanding* 139 (2015), 1–20.

T. Schöps, T. Sattler, and M. Pollefeys. 2020. SurfelMeshing: Online Surfel-Based Mesh Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 10 (2020), 2494–2507. https://doi.org/10.1109/TPAMI.2019.2947048

A. Segal, D. Haehnel, and S. Thrun. 2009. Generalized-ICP.. In *Robotics: Science and Systems*, Vol. 2. 435.

Y. Shi, K. Xu, M. Nießner, S. Rusinkiewicz, and T. Funkhouser. 2018. PlaneMatch: Patch Coplanarity Prediction for Robust RGB-D Reconstruction. In *Computer Vision – ECCV 2018*. Springer International Publishing, Cham, 767–784. https://doi.org/10.1007/978-3-030-01237-3_46

J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. 2012. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 573–580.

J. Stückler and S. Behnke. 2014. Multi-resolution Surfel Maps for Efficient Dense 3D Modeling and Tracking. *Journal of Visual Communication and Image Representation* 25, 1 (2014), 137 – 147. https://doi.org/10.1016/j.jvcir.2013.02.008

E. Sucar, S. Liu, J. Ortiz, and A. J. Davison. 2021. iMAP: Implicit Mapping and Positioning in Real-Time. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 6229–6238.

E. Sucar, K. Wada, and A. Davison. 2020. NodeSLAM: Neural Object Descriptors for Multi-View Shape Reconstruction. In *2020 International Conference on 3D Vision (3DV)*. 949–958. https://doi.org/10.1109/3DV50981.2020.00105

J. Sun, Y. Xie, L. Chen, X. Zhou, and H. Bao. 2021. NeuralRecon: Real-Time Coherent 3D Reconstruction From Monocular Video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 15598–15607.

C. Wang and X. Guo. 2017. Feature-based RGB-D camera pose optimization for real-time 3D reconstruction. *Computational Visual Media* 3, 2 (June 2017), 95–106. https://doi.org/10.1007/s41095-016-0072-2

O. Wasenmüller, M. Meyer, and D. Stricker. 2016. CoRBS: Comprehensive RGB-D Benchmark for SLAM Using Kinect v2. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 1–7. https://doi.org/10.1109/WACV.2016.7477636

S. Weder, J. Schönberger, M. Pollefeys, and M. R. Oswald. 2020. RoutedFusion: Learning Real-Time Depth Map Fusion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4886–4896. https://doi.org/10.1109/CVPR42600.2020.00494

S. Weder, J. Schönberger, M. Pollefeys, and M. R. Oswald. 2021. NeuralFusion: Online Depth Fusion in Latent Space. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3161–3171. https://doi.org/10.1109/CVPR46437.2021.00318

T. Weise, T. Wismer, B. Leibe, and L. Van Gool. 2009. In-hand Scanning with Online Loop Closure. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. 1630–1637. https://doi.org/10.1109/ICCVW.2009.5457479

H. Wendland. 1995. Piecewise Polynomial, Positive Definite and Compactly Supported Radial Functions of Minimal Degree. *Advances in Computational Mathematics* 4, 1 (01 Dec 1995), 389–396. https://doi.org/10.1007/BF02123482

T. Whelan, M. Kaess, M.F. Fallon, H. Johannsson, J.J. Leonard, and J.B. McDonald. 2012. Kintinuous: Spatially Extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*. Sydney, Australia.

T. Whelan, R. Salas-Moreno, B. Glocker, A. Davison, and S. Leutenegger. 2016. ElasticFusion: Real-time Dense SLAM and Light Source Estimation. *The International Journal of Robotics Research* 35, 14 (2016), 1697–1716. https://doi.org/10.1177/0278364916669237

S. Wu, W. Sun, P. Long, H. Huang, D. Cohen-Or, M. Gong, O. Deussen, and B. Chen. 2014. Quality-Driven Poisson-Guided Autoscanning. *ACM Trans. Graph.* 33, 6, Article 203 (Nov. 2014), 12 pages.

S. Yang, K. Chen, M. Liu, H. Fu, and S. Hu. 2017. Saliency-aware Real-time Volumetric Fusion for Object Reconstruction. *Computer Graphics Forum* 36, 7 (2017), 167–174. https://doi.org/10.1111/cgf.13282

S. Yang, B. Li, Y. Cao, H. Fu, Y. Lai, L. Kobbelt, and S. Hu. 2020. Noise-Resilient Reconstruction of Panoramas and 3D Scenes Using Robot-Mounted Unsynchronized Commodity RGB-D Cameras. *ACM Trans. Graph.* 39, 5, Article 152 (July 2020), 15 pages. https://doi.org/10.1145/3389412

C. Zhang and Y. Hu. 2017. CuFusion: Accurate Real-time Camera Tracking and Volumetric Scene Reconstruction with a Cuboid. *Sensors* 17, 10 (2017), 2260.

X. Zhang, H. Li, and Z. Cheng. 2008. Curvature estimation of 3D point cloud surfaces through the fitting of normal section curvatures. *Proceedings of AsiaGraph 2008* (01 2008), 72–79.

Y. Zhang, W. Xu, Y. Tong, and K. Zhou. 2015. Online Structure Analysis for Real-Time Indoor Scene Reconstruction. *ACM Trans. Graph.* 34, 5, Article 159 (Nov. 2015), 13 pages. https://doi.org/10.1145/2768821

Q. Zhou and V. Koltun. 2013. Dense Scene Reconstruction with Points of Interest. *ACM Trans. Graph.* 32, 4, Article 112 (July 2013), 8 pages. https://doi.org/10.1145/2461912.2461919

Q. Zhou and V. Koltun. 2015. Depth Camera Tracking with Contour Cues. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 632–638. https://doi.org/10.1109/CVPR.2015.7298662

Q. Zhou, S. Miller, and V. Koltun. 2013. Elastic Fragments for Dense Scene Reconstruction. In *2013 IEEE International Conference on Computer Vision (ICCV)*. 473–480. https://doi.org/10.1109/ICCV.2013.65