Real-time High-accuracy 3D Reconstruction with Consumer RGB-D Cameras

YAN-PEI CAO, Tsinghua University, P.R. China
LEIF KOBBELT, RWTH Aachen University, Germany
SHI-MIN HU*, Tsinghua University, P.R. China

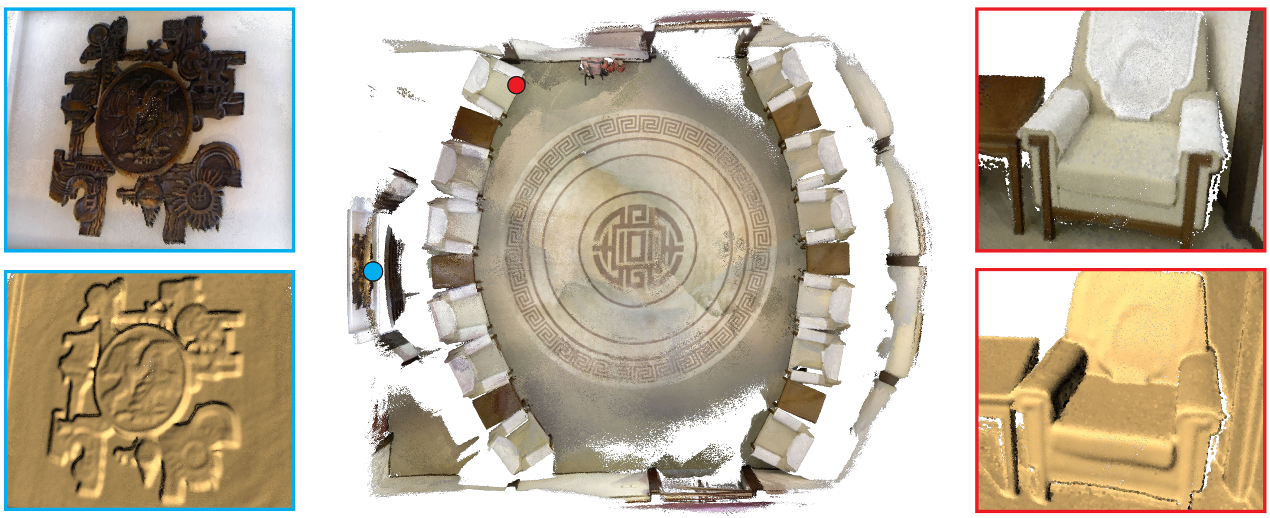


Fig. 1. We achieve real-time high-fidelity 3D reconstruction with consumer RGB-D cameras. The middle image shows the top view of a reconstructed lounge room. Close-ups show two accurately reconstructed details. Left: A wall relief (the top, left image shows captured texture, while the left, bottom image uses diffuse shading to better show the reconstructed geometry). Right: A lounge chair. This scene is scanned by an *Asus XTion PRO LIVE* sensor.

We present an integrated approach for reconstructing high-fidelity 3D models using consumer RGB-D cameras. RGB-D registration and reconstruction algorithms are prone to errors from scanning noise, making it hard to perform 3D reconstruction accurately. The key idea of our method is to assign a probabilistic uncertainty model to each depth measurement, which then guides the scan alignment and depth fusion. This allows us to effectively handle inherent noise and distortion in depth maps while keeping the overall scan registration procedure under the iterative closest point (ICP) framework for simplicity and efficiency. We further introduce a local-to-global, submap-based, and uncertainty-aware global pose optimization scheme to improve scalability and guarantee global model consistency. Finally, we have implemented the proposed algorithm on the GPU, achieving real-time 3D scanning frame rates and updating the reconstructed model on-the-fly. Experimental results on simulated and real-world data demonstrate that the proposed method outperforms state-of-the-art systems in terms of the accuracy of both recovered camera trajectories and reconstructed models.

CCS Concepts: • **Computing methodologies → Shape modeling**; *Computer graphics*;

Additional Key Words and Phrases: RGB-D scanning, 3D scan registration, scene reconstruction

ACM Reference Format:
Yan-Pei Cao, Leif Kobbelt, and Shi-Min Hu. 2018. Real-time High-accuracy 3D Reconstruction with Consumer RGB-D Cameras. *ACM Trans. Graph.* 1, 1 (May 2018), 16 pages. https://doi.org/0000001.0000001_2

1 INTRODUCTION

High-accuracy reconstruction of 3D objects and scenes is key to mixed reality applications and the next generation of robotics. The availability of consumer RGB-D cameras provides an opportunity for many users to access scanned 3D models, leading to a resurgence of research into RGB-D mapping and 3D reconstruction systems. In the robotics community, the problem of generating sparse or dense models of a static environment with mobile robots or handheld cameras is known as simultaneous localization and mapping (SLAM); in computer vision and graphics, since the pioneering *KinectFusion* work [Newcombe et al. 2011], many approaches, both real-time and offline, have been proposed to reconstruct scenes from depth cameras or RGB-D streams [Choi et al. 2015; Dai et al. 2017;

*Also with Cardiff University.

This work was supported by the Joint NSFC-DFG Research Program (Project Number 61761136018), the Natural Science Foundation of China (Project Number 61521002), the European Research Council, ERC Advanced Grant ACROSS (340884) and the German Research Foundation, DFG (Gottfried-Wilhelm-Leibniz-Program and the Excellence Initiative).
Authors' addresses: Yan-Pei Cao, Tsinghua University, Beijing, 100084, P.R. China, caoyanpei@gmail.com; Leif Kobbelt, RWTH Aachen University, Aachen, 52074, Germany, kobbelt@cs.rwth-aachen.de; Shi-Min Hu, Tsinghua University, Beijing, 100084, P.R. China, shimin@tsinghua.edu.cn.

2018. 0730-0301/2018/5-ART $15.00
https://doi.org/0000001.0000001_2

Kähler et al. 2016; Nießner et al. 2013; Wang and Guo 2017; Whelan et al. 2015b].

Despite recent advances in indoor scene reconstruction, acquiring high-fidelity 3D models with data streams from consumer-level RGB-D sensors is still a particularly challenging problem. Prior systems [Newcombe et al. 2011; Nießner et al. 2013; Wang et al. 2016] often use a volumetric representation of the scene geometry, the truncated signed distance function (TSDF) [Curless and Levoy 1996], which is beneficial for fast camera tracking and frame data integration. However, depth data acquired by consumer depth sensors always contains a significant amount of noise, making depth fusion within a voxel suffer from blurring or over-smoothed geometric details. Another and more important challenge for large environment scanning is tracking drift, i.e. accumulation of trajectory error over time, which can distort the reconstructed surfaces. Noisy input data makes frame tracking even more unreliable and further aggravates the problem of drift. Previous work has proposed a variety of global registration techniques to resolve inconsistent alignments [Choi et al. 2015; Zeng et al. 2016]; on the other hand, several probabilistic registration algorithms have been proposed to improve the accuracy and robustness of geometric alignment [Danelljan et al. 2016; Jian and Vemuri 2011]. These algorithms, however, are either designed for pairwise registration, or operate on a small set of point clouds jointly, and cannot handle depth streams directly.

In this work, we present a practical approach to high-fidelity reconstruction of 3D scenes using consumer RGB-D cameras, outperforming state-of-the-art automatic scene reconstruction methods regarding the accuracy of the recovered camera poses and reconstructed models. Recognizing that handling the inherent noise and distortion in depth maps is key to achieving accuracy in both registration and reconstruction, we apply a probabilistic uncertainty model to each depth measurement, which considers the measured points in a depth scan as samples generated by the surfaces in the scene, and reflects the physical properties of the underlying depth sensor. Unlike existing methods that try to compensate for depth noise in a preprocessing step, we estimate a pointwise uncertainty model using a measurement's local temporal neighborhood across nearby frames *on-the-fly*, which obliviously handles noise coming from viewing distance, obliqueness, surface material, depth discontinuity (silhouettes), and radial distance, and generalize the standard iterative closest point (ICP) algorithm [Besl and McKay 1992] by embedding the estimated probability distribution into the transformation estimation stage. This modified ICP model with spatial uncertainty handles noise and distortion well, thus significantly outperforms other ICP variants (e.g. point-to-plane ICP, plane-to-plane ICP, and sparse ICP) [Bouaziz et al. 2013; Rusinkiewicz and Levoy 2001; Segal et al. 2009]. Furthermore, in reconstruction, our uncertainty model guides the integration of depth points in a smarter manner, pulling noisy and distorted depth points back to the original surfaces.

In practice, scanning trajectories can be very complex, and 3D scenes can be geometrically and photometrically featureless, thus tracking drift cannot be fully avoided by using only local registration. We then present an uncertainty-aware RGB-D bundle adjustment (BA) scheme to achieve global model consistency. To process thousands of input frames efficiently, we split input sequences into submaps, and only perform BA at the level of submaps. Finally, we have implemented the proposed algorithm on the GPU, achieving real-time 3D scanning frame rates and making it ready for use in practical applications.

We evaluate the proposed framework on both simulated and real-world datasets and show that our approach increases the accuracy of camera trajectories and reconstructed models.

In summary, our work makes three contributions. Firstly, we develop a practical RGB-D reconstruction framework, which takes into account the uncertainty in depth measurements from consumer RGB-D cameras and achieves real-time high-accuracy reconstruction results for 3D objects and scenes while scaling well for large environments by use of a local-to-global optimization scheme. Secondly, we provide a new variant of the classical ICP algorithm. Exploiting the spatial uncertainty information, it outperforms existing approaches for aligning noisy depth scans. Thirdly, we give an uncertainty-guided RGB-D integration method, which reduces noise while preserving geometric features of the underlying surface structures.

2 RELATED WORK

The pioneering KinectFusion system has stimulated research into 3D reconstruction and shape acquisition with handheld RGB-D cameras. The core of this system is an implicit 3D volumetric scene representation base on TSDF into which each new input frame is registered and fused. A fundamental limitation is its lack of scalability, as the uniform voxel grid used is memory-intensive. This can be overcome by exploiting the sparsity of TSDF grids to design hierarchical spatial subdivision strategies with efficient data structures [Chen et al. 2013; Nießner et al. 2013; Reichl et al. 2016; Roth and Vona 2012; Zeng et al. 2013]. Another important issue is that, although the frame-to-model registration for pose estimation is locally accurate, tracking drift cannot be avoided and accumulates to such a degree as to break the reconstruction.

In the robotics community, the issue of tracking drift has been extensively studied in the context of SLAM. Techniques that build dense maps of the environment are known as *dense* or *direct* methods [Endres et al. 2014; Engel et al. 2014; Henry et al. 2012; Kerl et al. 2013; Kümmerle et al. 2011; Mur-Artal et al. 2015; Steinbrücker et al. 2013]. These approaches either apply global pose graph optimization or bundle adjustment to get a globally consistent map, distributing the accumulated drift error across the graph. Loop closures are detected by matching individual frames with sparse visual features or fern encoding databases [Glocker et al. 2015]. Nevertheless, these methods focus on achieving highly accurate camera tracking in real-time and do not guarantee detailed geometry for the reconstructed 3D models.

In the works of [Li et al. 2013; Ruhnke et al. 2012; Whelan et al. 2015a,b; Zhou et al. 2013], global consistency is achieved in the opposite way. The reconstructed 3D models are deformed globally [Li et al. 2013; Ruhnke et al. 2012; Whelan et al. 2015a,b] or locally [Zhou et al. 2013] to meet loop closure or local constraints. Amongst these methods, [Whelan et al. 2015a] and [Whelan et al. 2015b] can handle input streams at interactive frame-rates, while [Li et al. 2013; Ruhnke et al. 2012; Zhou et al. 2013] run offline. [Li et al. 2013] is specially designed for human scanning. However, the frequent deformation in these methods leads to high computational cost.

Another alternative is to use submaps, as proposed in [Choi et al. 2015; Dai et al. 2017; Fioraio et al. 2015; Kähler et al. 2016; Maier et al. 2014]. Compared to using a single global map, local submaps are less affected by drift error and better preserve detailed shape [Choi et al. 2015; Fioraio et al. 2015]. Furthermore, submaps can reduce computational costs and memory consumption, thus making the optimization more efficient [Dai et al. 2017; Kähler et al. 2016; Maier et al. 2014]. In [Maier et al. 2014], submaps are internally optimized by bundle adjustment. [Choi et al. 2015; Fioraio et al. 2015] use dense geometric correspondences to align submaps globally, while [Dai et al. 2017] uses both sparse and dense correspondences from RGB and depth channels. [Kähler et al. 2016] estimates constraints between submaps by tracking within multiple submaps simultaneously and detects loop closure explicitly. Amongst above methods, the system presented in [Dai et al. 2017] achieves the state-of-the-art reconstruction quality. Yet, it requires *two* high-end GPUs to run in real-time, limiting its practical use.

Relative pose estimation (either frame-to-frame or frame-to-model) usually uses geometric registration algorithms, such as variants of the ICP algorithm [Rusinkiewicz and Levoy 2001]. They can produce tight alignment when well initialized, but are unreliable without such initialization or when presented with noisy data. Thus, many global registration techniques have been proposed to deal with wide baseline matching. [Mellado et al. 2014] iteratively estimates alignment from sparse correspondences and then validates the result with full correspondences; [Yang et al. 2013] explores the pose space and searches for the optimal alignment, which theoretically guarantees optimality but is rather time-consuming; [Zhou et al. 2016] presents a fast global registration technique which does not require initialization, iterative sampling and validation, or local refinement. These approaches, however, are not reliable enough when a significant amount of noise is presented in input scans.

In order to handle noisy data, [Jian and Vemuri 2011; Myronenko and Song 2010] employ mixture models to depict the distributions of input point clouds and formulate the point sets registration problem as solving an alignment between the distribution of two point sets. Specifically, the Coherent Point Drift (CPD) [Myronenko and Song 2010] algorithm uses the Gaussian Mixture Model (GMM) to describe the distribution of the *template* set and seeks the optimal (rigid or non-rigid) transformation between two input sets by maximizing the likelihood function, which in essence is minimizing the Kullback-Leibler divergence between a GMM (the *template*) and a mixture of Dirac delta functions (the *data*); while in [Jian and Vemuri 2011], both template and data point sets are modeled by GMMs, and the two point sets are aligned by minimizing the ℓ_2 distance between two corresponding GMMs, which leads to a closed-form solution. Interestingly, the general probabilistic point set registration framework proposed in [Jian and Vemuri 2011] can interpret several previous registration algorithms as particular cases. Recently, [Danelljan et al. 2016] proposes a probabilistic registration technique which incorporates color information. However, these probabilistic registration algorithms often need a long time to find the optimal alignment, and so are not suitable for RGB-D reconstruction with thousands of input scans.

In [Cui et al. 2013], a noise-resilient but *offline* approach is proposed for 3D object scanning, which suppresses random data noise by splitting the input sequence into batches and computing superresolved depth scans within each batch with a nonlinear anisotropic regularizer; if input scans are captured by a time-of-flight (ToF) camera which brings systematic biases, it then adopts the CPD algorithm for probabilistic scan alignment, where the transformation of data points is parameterized by a rigid motion and a non-rigid warp along the viewing direction (representing measurement biases). If the input scans are already aligned or a rough surface model is available, the Bayesian paradigm can be employed to achieve noise-tolerant surface reconstruction [Diebel et al. 2006; Jenke et al. 2006; Paulsen et al. 2010]. The Bayesian framework intuitively breaks down the noise-aware reconstruction problem into a probabilistic measurement model and priors about the scene geometry. [Jenke et al. 2006] reduces the noise in the input and generates an upsampled point cloud, which takes a mixture of truncated Gaussians and a uniform distribution, as the measurement model while including both smoothness and density in the prior. Discrete attributes, such as point types (i.e., region, edge, or corner), are also assigned to each point to preserve sharp features explicitly. Similarly, [Diebel et al. 2006] presents a Bayesian-based approach to regularizing and decimating surface models, which works solely with mesh representations as the definition of its smoothness prior requires mesh topology. Furthermore, [Paulsen et al. 2010] chooses to define the Markov Random Field regularizer on an implicit representation of the surface (i.e., the signed distance field). While theoretically appealing, these statistical approaches are too time-consuming for real-time applications.

Our approach is based on an uncertainty model for measurements of consumer-grade RGB-D cameras. [Grigoryan and Rheingans 2004; Kalaiah and Varshney 2003] introduce statistical and uncertainty information into the point cloud representation, but the uncertainty information is only used for compression and visualization. [Fuhrmann and Goesele 2014] derive the *sample scale*, which can also be interpreted as the spatial sampling uncertainty on the object surface, from the viewing distance of the sensor, but only employs this sampling uncertainty during the surface reconstruction process. On the other hand, previous noise models for RGB-D cameras are either designed for a specific type of sensor [Ferstl et al. 2015; Jordt and Koch 2013; Lenzen et al. 2013; Nguyen et al. 2012; Park et al. 2012; Reynolds et al. 2011; Yamazoe et al. 2018] or require sensor calibration [Kim et al. 2008; Teichman et al. 2013; Wang et al. 2016]. Moreover, these models are proposed to compensate for noise or undistort raw depth maps as a preprocessing step. In contrast, we present a simple yet general uncertainty model, which can be estimated in the process of reconstruction and thus is free of pre-calibration of the camera. Besides, the presented uncertainty model is used to guide the registration and integration of input scans, instead of pre-filtering the input data before the reconstruction starts.

3 UNCERTAINTY-AWARE RGB-D SCAN REGISTRATION AND INTEGRATION

Frame-to-frame or frame-to-model pose estimation (or, tracking) is an essential component of most RGB-D reconstruction systems. Although several alternatives [Mellado et al. 2014; Zhou et al. 2016] have been proposed in recent years, due to their simplicity and efficiency, ICP and its variants (especially point-to-plane ICP) are still the most widely used techniques for scan alignment. However, in practice, these techniques are prone to noise and distortion, making tracking highly brittle. Thus, we first introduce a novel and general uncertainty model for depth scans acquired by consumer-level RGB-D sensors (Section 3.1), then, based on the uncertainty model, we propose an uncertainty-aware, frame-to-model RGB-D registration and integration algorithm (Section 3.2), which tolerates noise and distortion in input scans while maintaining the overall computational pipeline under the framework of ICP.

For the rest of the paper, we use transformation matrices $\mathbf{T}_n \in SE(3)$, $n = 1 \dots N$ to represent camera poses, where

$$\mathbf{T}_n = \begin{pmatrix} \mathbf{R}_n & \mathbf{t}_n \\ \mathbf{0} & 1 \end{pmatrix}, \tag{1}$$

with rotation $\mathbf{R}_n \in SO(3)$ and translation $\mathbf{t}_n \in \mathbb{R}^3$. We then use $\boldsymbol{\mu}_{nk} \in \mathbb{R}^3$ to denote 3D surface point positions, with $k = 1 \dots K$, where $\boldsymbol{\mu}_{nk}$ denotes the k-th surface point acquired in the n-th camera pose (i.e. n-th frame), expressed in the global coordinate frame. Similarly, $\mathbf{p}_{nk} \in \mathbb{R}^3$ represents the k-th noisy 3D measurement acquired in the n-th camera pose. For notation simplicity, conversion between 3D vectors and their corresponding 4D homogeneous vectors is omitted. Input RGB-D scans are denoted by $\{\mathcal{F}_n = \{C_n, \mathcal{D}_n\}\}$, where C_n is the RGB image and $\mathcal{D}_n$ is the depth map.

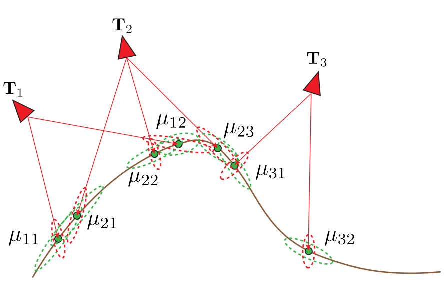


Fig. 2. Uncertainty model used in our algorithm. Red triangles: camera poses. Large green dots: surface points. Green/red dashed ellipses indicate the surface model (see Section 3.1.1) and measurement uncertainty model (see Section 3.1.2), respectively.

3.1 Uncertainty Modeling

We start by introducing our uncertainty model for depth scans. For each acquired depth sample, we assume there are two different types of uncertainty sources that can make it stray from its true 3D position, i.e., the surface sampling uncertainty (Section 3.1.1) and the measurement uncertainty (Section 3.1.2).

3.1.1 Surface Sampling Uncertainty. Due to the limited spatial resolution of consumer-level RGB-D sensors, we may not be able to sample the exact same surface point from different scans. Inspired by [Segal et al. 2009], we assume surfaces in 3D scenes are piecewise smooth, and we describe the local surface characteristics of each measured point using normal distributions. More specifically, we assume that each measurement $\mathbf{p}_{nk}$ is generated from a surface point $\boldsymbol{\mu}_{nk}$ by a normal distribution $\mathcal{N}(\boldsymbol{\mu}_{nk}, \Sigma_{nk}^{\text{surf}})$. Here, $\Sigma_{nk}^{\text{surf}}$ is the covariance matrix estimated from the local neighborhood of $\mathbf{p}_{nk}$. The three eigenvectors of $\Sigma_{nk}^{\text{surf}}$, which are computed using the PCA, define a local frame at $\mathbf{p}_{nk}$; the eigenvector corresponding to the smallest eigenvalue indicates the normal direction, while the other two span the tangent plane. This implies that each measured point is distributed with low variance along its normal direction and relatively higher variance within its tangent plane. Green dashed ellipses in Fig. 2 depict the above surface uncertainty model.

3.1.2 Measurement Uncertainty. Besides the uncertainty from surface sampling, a more important source of uncertainty comes from the measurement noise in depth scans. Several noise models for consumer RGB-D cameras have already been proposed [Jordt and Koch 2013; Nguyen et al. 2012; Park et al. 2012; Reynolds et al. 2011; Teichman et al. 2013]. Currently, there are two different types of consumer-level depth sensors, i.e., active stereo sensors (e.g., *Microsoft Kinect v1* and other *PrimeSense*-derived sensors) and ToF sensors (e.g., *Microsoft Kinect v2*), and they hold very different noise properties. Apart from random noise, data captured by ToF cameras often exhibit a certain amount of pixel-dependent depth biases [Cui et al. 2013; Jordt and Koch 2013; Kim et al. 2008], which may depend on many factors, such as distance, material reflectance properties, surface orientation, etc. [Reynolds et al. 2011] regresses a per-pixel confidence map for ToF data using a Random Forest trained with real-world data and analyzes the contributions of different factors to the confidence model; [Jordt and Koch 2013] employs a polynomial interpolated Gaussian distribution with respect to depth and image amplitude as the noise model; while [Ferstl et al. 2015] exploits information from both depth and color channels for camera calibration, and similar to [Reynolds et al. 2011], it learns a mapping function between the captured depth and image intensity to the offset from the ground-truth depth value using a Random Forest. On the other hand, for active stereo depth cameras, noise and outliers are usually caused by limitations of stereo matching algorithms, such as quantization and edge fattening (due to false matches along depth boundaries). These errors are rather irregular and thus harder to model. [Nguyen et al. 2012] fits axial and lateral noise models of the Kinect v1 sensor as functions of the viewing distance and angle, applying the resulting noise models to the KinectFusion system for higher tracking accuracy; [Yamazoe et al. 2018] considers both camera and *projector* distortion, calibrating the Microsoft Kinect sensor using a moving chessboard; while [Teichman et al. 2013] presents an iterative approach to intrinsic calibration of PrimeSense-derived depth sensors, which integrates a SLAM system and utilizes the relatively accurate short-range measurements to estimate the low-frequency warping function for long-range measurements. The above approaches, however, either need tedious camera calibration beforehand or may only work with a certain type of sensor. Therefore, in this work, we present a simple yet general measurement uncertainty model, which needs no pre-calibration of the camera. The uncertainty introduced by the depth measurement of a 3D point is also represented by a normal distribution. Note that we assume there is no correlation of noise between the axis directions in image space, so the covariance matrix $\Sigma^{\text{meas}}_{uvd}$ in image space has diagonal form:

$$\Sigma^{\text{meas}}_{uvd} = \begin{pmatrix} \sigma_u^2 & 0 & 0 \\ 0 & \sigma_v^2 & 0 \\ 0 & 0 & \sigma_d^2 \end{pmatrix}, \tag{2}$$

where σ_u^2 and σ_v^2 represent the variance in point position in the two coordinate directions in the image plane, caused by pixel quantization or keypoint localization, and σ_d^2 is the variance in depth measurements, resulting from different radial distance, measuring distance, material of the object surface, etc. Furthermore, given a camera model function $\mathbf{f}(u, v, d)$ which maps points in the image space to the camera space, we can obtain the covariance matrix Σ^{meas} in the camera space:

$$\Sigma^{\text{meas}} = \mathbf{J_f} \cdot \Sigma^{\text{meas}}_{uvd} \cdot \mathbf{J_f}^{\top}, \tag{3}$$

where $\mathbf{J_f}$ is the Jacobian of $\mathbf{f}(u, v, d)$. Red dashed ellipses in Fig. 2 illustrate the measurement uncertainty.

We propose a novel approach to estimate the depth variance. As has been observed previously [Choi et al. 2015; Zhou et al. 2013], although geometric registration methods (e.g. point-to-plane ICP) are unreliable over long ranges, they are quite accurate locally. Given a sequence of depth scans, we take B consecutive frames as a batch $\mathcal{B}$, set frame $\lceil B/2 \rceil$ as the central frame $\mathcal{D}_{\lceil B/2 \rceil}$, and compute local alignments from other frames in $\mathcal{B}$ to the central frame $\mathcal{D}_{\lceil B/2 \rceil}$ using point-to-plane ICP. Note that each pairwise alignment only needs to be computed once and can be used in later steps for final registration and integration (see Section 3.2), so there is no additional computational overhead. Then we use these *local* poses to re-project each aligned point cloud in $\mathcal{B}$ into a depth map seen from the *local* pose of $\mathcal{D}_{\lceil B/2 \rceil}$, and calculate a depth variance map $\mathcal{V}_{\lceil B/2 \rceil}$. Each entry of $\mathcal{V}_{\lceil B/2 \rceil}$ is the depth variance of the corresponding entry in depth scan $\mathcal{D}_{\lceil B/2 \rceil}$. For σ_u^2 and σ_v^2, if considering dense correspondences (Section 3.2), they are set to be 0.5^2, i.e., the standard deviations are half pixel size; otherwise, if sparse feature matching is used (Section 4.3), we apply the constant values estimated by [Park et al. 2012], which capture the worst-case variances of 2D keypoint locations for image keypoint detectors. Fig. 3 illustrates the log-magnitude of the estimated measurement uncertainty of an example RGB-D frame. Results of using three different batch sizes ($B = 11, 21, 41$) are shown. We find the reconstruction quality of our system is not sensitive to the value of B and set $B = 21$ in our experiments.

One natural byproduct of the above process is that we can perform a *temporal* median filtering to the central depth frame $\mathcal{D}_{\lceil B/2 \rceil}$, i.e., replacing $\mathcal{D}_{\lceil B/2 \rceil}$ with the median depth map $\tilde{\mathcal{D}}_{\lceil B/2 \rceil}$ computed in the batch $\mathcal{B}$. As shown in Fig. 4, this filtering step can effectively reduce both low-frequency and high-frequency noise in depth data *captured by different types of sensors.*

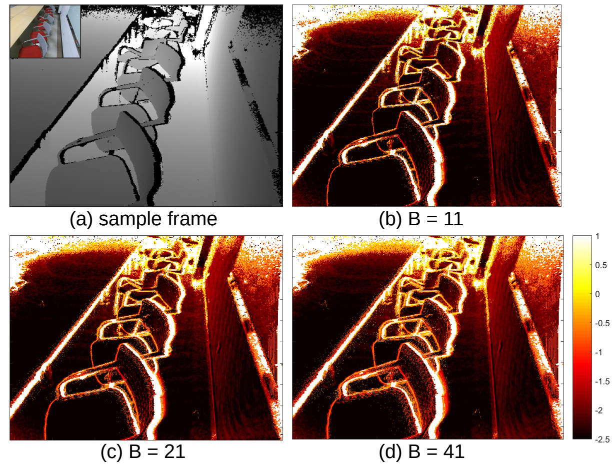


Fig. 3. (a): A depth image; corresponding RGB image shown in the inset. (b)-(d): Log-magnitude of measurement uncertainty calculated using three different batch sizes B. This frame is captured by a *Microsoft Kinect v2*[1] sensor.

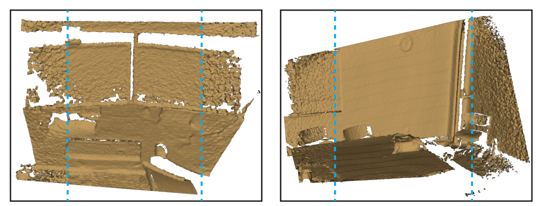


Fig. 4. Point clouds generated from depth maps. The filtered result is overlaid in the middle of each image. Left: Data captured by a *Structure Sensor*[2]. Right: Data captured by a *Microsoft Kinect v2* sensor.

3.2 Uncertainty-aware Frame-to-model Registration and Integration

Using the previously introduced surface and measurement uncertainty models, we now assume each 3D measurement $\mathbf{p}_{nk}$ is generated by the underlying 3D surface point $\boldsymbol{\mu}_{nk}$ according to the Gaussian mixture

$$\mathbf{p}_{nk} \sim \phi^{\text{surf}} \mathcal{N}(\boldsymbol{\mu}_{nk}, \Sigma^{\text{surf}}_{nk}) + \phi^{\text{meas}} \mathcal{N}(\boldsymbol{\mu}_{nk}, \Sigma^{\text{meas}}_{nk}), \tag{4}$$

where ϕ^{surf} and ϕ^{meas} are mixture weights. We use $\phi^{\text{surf}} = \phi^{\text{meas}} = 0.5$ in our paper, and for notation simplicity, they are omitted in the equations for the rest of the paper.

3.2.1 Frame-to-frame Alignment. Between a given RGB-D scan $\mathcal{F}_s$ and a target scan $\mathcal{F}_t$ we aim to find the transformation $\mathbf{T}^*$ that aligns two scans best, where $\mathbf{T}^*\mathbf{p} = \mathbf{R}^*\mathbf{p} + \mathbf{t}^*$. Assuming $\mathbf{p}_{sk}$ and $\mathbf{p}_{tk}$ are two corresponding samples in $\mathcal{F}_s$ and $\mathcal{F}_t$, and according to our uncertainty model (Eq. 4), they are drawn from two independent Gaussian mixtures, so the distribution of the offset vector $\Delta\mathbf{p}_k(\mathbf{T}^*) = \mathbf{p}_{tk} - \mathbf{T}^*\mathbf{p}_{sk}$ can be derived as

$$\begin{aligned} \Delta\mathbf{p}_k(\mathbf{T}^*) &\sim \sum_{m=1}^{M} \mathcal{N}(\boldsymbol{\mu}_{tk} - \mathbf{T}^*\boldsymbol{\mu}_{sk}, \Sigma_{k_m}) \\ &= \sum_{m=1}^{M} \mathcal{N}(\mathbf{0}, \Sigma_{k_m}), \end{aligned} \tag{5}$$

where M is the number of *modes* in the Gaussian mixture (here $M = 4$), $\Sigma_{k_1} = \Sigma_{tk}^{\text{surf}} + \mathbf{R}^*\Sigma_{sk}^{\text{surf}}\mathbf{R}^{*\top}$, $\Sigma_{k_2} = \Sigma_{tk}^{\text{surf}} + \mathbf{R}^*\Sigma_{sk}^{\text{meas}}\mathbf{R}^{*\top}$, $\Sigma_{k_3} = \Sigma_{tk}^{\text{meas}} + \mathbf{R}^*\Sigma_{sk}^{\text{surf}}\mathbf{R}^{*\top}$, and $\Sigma_{k_4} = \Sigma_{tk}^{\text{meas}} + \mathbf{R}^*\Sigma_{sk}^{\text{meas}}\mathbf{R}^{*\top}$. The second step is based on the assumption that $\mathbf{T}^*$ perfectly aligns $\boldsymbol{\mu}_{sk}$ to $\boldsymbol{\mu}_{tk}$ ([Segal et al. 2009]), i.e., $\boldsymbol{\mu}_{tk} = \mathbf{T}^*\boldsymbol{\mu}_{sk}$.

Now $\mathbf{T}^*$ can be computed using maximum likelihood estimation (MLE). However, the maximum likelihood solution is quite complicated: the logarithm operator cannot be pushed inside the sum mixture. So we apply the max-mixture approximation [Olson and Agarwal 2013]:

$$\begin{aligned}
\mathbf{T}^* &= \underset{\mathbf{T}}{\operatorname{argmax}} \prod_{k\in C} prob\left(\Delta\mathbf{p}_k\left(\mathbf{T}\right)\right) \\
&= \underset{\mathbf{T}}{\operatorname{argmax}} \prod_{k\in C} \sum_{m\in M} \mathcal{N}(\Delta\mathbf{p}_k\left(\mathbf{T}\right), \Sigma_{k_m}) \\
&\approx \underset{\mathbf{T}}{\operatorname{argmax}} \prod_{k\in C} \max_{m\in M} \mathcal{N}(\Delta\mathbf{p}_k\left(\mathbf{T}\right), \Sigma_{k_m}) \\
&= \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{k\in C} \min_{m\in M} \left(\frac{1}{2}\Delta\mathbf{p}_k\left(\mathbf{T}\right)^\top \Sigma_{k_m}^{-1} \Delta\mathbf{p}_k\left(\mathbf{T}\right) - \log(\phi_{k_m}\eta_{k_m})\right),
\end{aligned} \tag{6}$$

where ϕ_{k_m} and η_{k_m} are the mixture weight and the normalization weight, respectively, and C is the set of correspondences. We also define

$$E_{geo} = \frac{1}{|C|} \sum_{k\in C} \min_{m\in M} \left(\frac{1}{2}\Delta\mathbf{p}_k\left(\mathbf{T}\right)^\top \Sigma_{k_m}^{-1} \Delta\mathbf{p}_k\left(\mathbf{T}\right) - \log(\phi_{k_m}\eta_{k_m})\right). \tag{7}$$

The uncertainty information about the positions of 3D measurements encoded in $\{\Sigma_{k_m}\}$ makes E_{geo} robust to measurement noise. Here, if we only consider the surface sampling uncertainty, E_{geo} degenerates to the plane-to-plane case in [Segal et al. 2009].

Since RGB-D scans also contain color information, we additionally use dense photometric constraints for alignment:

$$E_{rgb} = \frac{1}{|\chi_s|} \sum_{k\in\chi_s} \left(\mathcal{I}_t\left(\boldsymbol{\pi}(\mathbf{T}\mathbf{p}_{sk})\right) - \mathcal{I}_s\left(\boldsymbol{\pi}(\mathbf{p}_{sk})\right)\right)^2. \tag{8}$$

Here, $\boldsymbol{\pi}$ is the projection function, and χ_s denotes the set of valid samples in frame $\mathcal{F}_s$. $\mathcal{I}_s$ and $\mathcal{I}_t$ are intensity images computed from C_s and C_t.

Now we wish to minimize both geometric and photometric error, based on the following energy:

$$E_{rgbd} = w_{geo}E_{geo} + E_{rgb}, \tag{9}$$

where w_{geo} is the weight of the geometric term, and we find $w_{geo} = 4$ works well in our experiments. Minimizing the energy in Eq. 9 leads to a nonlinear least squares problem, which is solved by the Gauss-Newton algorithm (more detail in Section 4.2). Pairwise transformations calculated by point-to-plane ICP for estimating the measurement uncertainty (see Section 3.1.2) are used to initialize the optimization, speeding up convergence.

Correspondence search. Correspondences are crucial to correct convergence of the ICP algorithm. Here, correspondences are still determined using Euclidean distance for *fast* closest point look up. Given an initial set of candidate correspondences computed within a search radius r (we use $r = 0.2$ m in our experiments), we further apply the following correspondence filtering steps:

Uncertainty filter. Although our uncertainty-aware geometric error (Eq. 7) is designed to handle noise and distortion, correspondences computed using Euclidean neighborhoods are still prone to outliers. Thus, correspondences with high measurement uncertainty should be removed. To do so, we set a threshold σ^2 for the depth variance (σ_d^2 in Eq. 2) for the points in the correspondence set. If the depth variance of both ends of an associated corresponding pair exceeds σ^2, this correspondence is rejected. We find that $\sigma = 0.0075$ m works well in our experiments.

Normal filter. Corresponding points should have similar normal orientations. Therefore, if the normals of a pair of points differ by more than a threshold β (= 30°), we reject the correspondence.

Planar weighting. Measurements located in planar regions are typically more reliable than edge and corner measurements: tangents of points near sharp corners are not well defined, in which case the surface model Σ_{nk}^{surf} may not describe the measurement distribution properly. Thus, we define a planar weight for each measurement: $w_{nk}^p = 1 - 3|\lambda_0|/(|\lambda_0| + |\lambda_1| + |\lambda_2|)$, where $\{\lambda_i\}, i = 1, 2, 3$ are the eigenvalues of Σ_{nk}^{surf}, with $|\lambda_0| < |\lambda_1| < |\lambda_2|$. The weight of a correspondence pair $\langle\mathbf{p}_{sk}, \mathbf{p}_{tk}\rangle$ is then defined by $w_{sk}^p w_{tk}^p$, which is applied to each correspondence in Eq. 7.

3.2.2 Frame-to-frame Integration. Having computed frame transformations, we then integrate input scans into a unified global model to reduce the memory cost and further remove the noise. We start from the case of two scans in which $\mathcal{F}_s$ is integrated into $\mathcal{F}_t$. We wish to merge a pair of matched sample points $\langle\mathbf{p}_{sk}, \mathbf{p}_{tk}\rangle$ into a single point $\hat{\mathbf{p}}_{tk}$, whose position is consistent with the measurements $\langle\mathbf{p}_{sk}, \mathbf{p}_{tk}\rangle$, and conforms to their uncertainty model. Specifically, we seek the position of each integrated point $\hat{\mathbf{p}}_{tk}$ independently, by minimizing the following quadratic energy:

$$c_{sk}\|\mathbf{T}^{*-1}\hat{\mathbf{p}}_{tk} - \mathbf{p}_{sk}\|_{\Omega_{sk}}^2 + c_{tk}\|\hat{\mathbf{p}}_{tk} - \mathbf{p}_{tk}\|_{\Omega_{tk}}^2. \tag{10}$$

Here $\|\mathbf{x}\|_\Omega^2 := \mathbf{x}^\top\Omega^{-1}\mathbf{x}$ is the squared Mahalanobis distance with covariance matrix Ω, with $\Omega_{sk}^{-1} = \Sigma_{sk}^{\text{surf}-1} + \Sigma_{sk}^{\text{meas}-1}$ and $\Omega_{tk}^{-1} = \Sigma_{tk}^{\text{surf}-1} + \Sigma_{tk}^{\text{meas}-1}$. We define the *confidence weight* c_{nk} of a measurement $\mathbf{p}_{nk}$ to be $c_{nk} = \sigma/\sigma_{d,nk}$, where σ is the threshold for the standard deviation of depth (see Section 3.2.1), and $\sigma_{d,nk}$ is the standard deviation of the depth corresponding to $\mathbf{p}_{nk}$.

After integration, we assign the measurement uncertainty information of both $\mathbf{p}_{sk}$ and $\mathbf{p}_{tk}$ to $\hat{\mathbf{p}}_{tk}$, and re-estimate the surface sampling uncertainty of $\hat{\mathbf{p}}_{tk}$ using new neighboring points; i.e., $\hat{\mathbf{p}}_{tk}$ is drawn from a mixture of three modes:

$$\begin{aligned}
\hat{\mathbf{p}}_{tk} \sim \phi^{\text{surf}}\mathcal{N}(\hat{\boldsymbol{\mu}}_{tk}, \hat{\Sigma}_{tk}^{\text{surf}}) &+ \phi_{tk}^{\text{meas}}\mathcal{N}(\hat{\boldsymbol{\mu}}_{tk}, \Sigma_{tk}^{\text{meas}}) \\
&+ \phi_{sk}^{\text{meas}}\mathcal{N}(\hat{\boldsymbol{\mu}}_{tk}, \mathbf{R}^*\Sigma_{sk}^{\text{meas}}\mathbf{R}^{*\top}),
\end{aligned} \tag{11}$$

where $\phi_{tk}^{\text{meas}} = 0.5c_{tk}/(c_{tk} + c_{sk})$, $\phi_{sk}^{\text{meas}} = 0.5c_{sk}/(c_{tk} + c_{sk})$, $\phi^{\text{surf}} = 0.5$, and $\hat{\Sigma}_{tk}^{\text{surf}}$ is the re-estimated surface uncertainty covariance. Note that the correspondences used in the scan integration step are also filtered by measurement uncertainty and normal orientations, and the positions of unmatched measurements remain unchanged.

3.2.3 Frame-to-model Registration and Integration Scheme. Frame-to-model tracking usually provides more accurate results than the frame-to-frame approach [Newcombe et al. 2011], and our frame-to-frame uncertainty-aware registration algorithm (Section 3.2.1) can be easily adapted to the frame-to-model framework, as the integrated sample points (Section 3.2.2) are equipped with the same form of the uncertainty model as raw measurements. More specifically, by replacing the distribution of a raw measurement $\mathbf{p}_{nk}$ (two modes) with the distribution of an integrated global model point $\hat{\mathbf{p}}_{nk}$ (three or more modes), the only change in Eq. 5 and 6 is the number of Gaussian modes M, and the form of E_{geo} remains unchanged. However, the growth of the number of modes in the global model can lead to increasing computational and memory cost. Thus, we keep at most *three measurement* Gaussian components for each global model point, which are those having the lowest depth variance values.

4 HIERARCHICAL LARGE-SCALE RECONSTRUCTION WITH SUBMAPPING

Utilizing the previously presented uncertainty-aware frame-to-model tracking and integration algorithm (Section 3.2), for short sequences, we can elegantly handle input noise and obtain accurate camera trajectories and reconstruction results. However, in real-world situations, scanning sequences can contain thousands of frames, with very complex camera trajectories, and possibly visiting featureless areas, thus tracking drift cannot be fully avoided by using only frame-to-model registration. In order to ensure global model consistency while maintaining efficiency, we first split input sequences into submaps, which are individually reconstructed using the uncertainty-aware frame-to-model registration algorithm; then, inspired by previous SLAM systems [Maier et al. 2014; Mur-Artal et al. 2015], we further perform a local-to-global uncertainty-aware RGB-D bundle adjustment at the level of submaps.

4.1 Scene Representation

We adopt the surfel-based scene representation from [Keller et al. 2013] for large-scale reconstruction. Each surfel P_k is associated with the following attributes; a position $\mathbf{p}_k \in \mathbb{R}^3$, color $\mathbf{c}_k \in \mathbb{N}^3$, normal $\mathbf{n}_k \in \mathbb{R}^3$, 2-tuple $\boldsymbol{\psi}_{k_m} = \{\phi_{k_m}, \Sigma_{k_m}\}$ for each uncertainty mode m, radius $r_k \in \mathbb{R}$, weight $w_k \in \mathbb{R}$, and timestamp $t_k \in \mathbb{N}$. Here ϕ_{k_m} and Σ_{k_m} are the weight and covariance matrix of the m-th Gaussian component of the uncertainty model, respectively; other attributes are initialized following the same approach described in [Keller et al. 2013].

4.2 Submap Construction

Most previous methods create submaps based on a uniform partitioning scheme [Choi et al. 2015; Dai et al. 2017; Maier et al. 2014]: after every H frames, a new submap $\mathcal{S}$ is constructed. This simple criterion does not take into account camera motion behavior, and furthermore, the number of submaps increases linearly with the number of input frames. Together, these can produce redundant submaps or cause intra-submap drift. Instead, as in [Stückler and Behnke 2014], we construct new submaps based on the actual camera motion. If the rotation angle relative to the first frame of the current submap exceeds a threshold θ, or the relative translation exceeds δ, we start a new submap. Also, an additional threshold ϵ for the root-mean-squared distance (RMSD) between correspondences in frame-to-model registration is used to prevent tracking failure. We use $\theta = 20°$, $\delta = 0.3$ m, and $\epsilon = 0.01$ m in our experiments.

Implementation. To achieve real-time processing rates, the construction of submaps (i.e., the uncertainty-aware registration and integration) is performed fully on the GPU. The global model and its attributes are stored in an OpenGL texture. Note that we recover surface uncertainty covariance matrices from corresponding eigenvalues and normal vectors, and similarly, recover measurement uncertainty covariance matrices from depth variance and camera space coordinates (see Section 3.1.2), both on-the-fly, which can reduce the memory cost. Using the current estimate of the transformation $\mathbf{T}$ from the current scan to the global model, we predict a depth map and an RGB image from the global model via color splatted rendering and establish correspondences using projective data association [Newcombe et al. 2011]. Then we minimize E_{rgbd} (Eq. 9) using the Gauss-Newton algorithm, where $\{\Sigma_{k_m}\}$ in Eq. 7 are fixed in the linearization step and are then updated when a new estimate of the current transformation becomes available. Similar to [Keller et al. 2013; Newcombe et al. 2011; Whelan et al. 2015b], the Jacobian and residual are assembled and solved using CUDA, and multi-scale vertex and normal map pyramids are used. During scan integration, $\mathbf{p}_k$ and $\boldsymbol{\psi}_{k_m}$ are updated using our uncertainty-aware integration method (i.e., Eq. 10 and 11), while other attributes of P_k are updated in the same manner as described in [Keller et al. 2013]. Note the uncertainty estimation and depth filtering (Section 3.1.2) are also performed on the GPU.

4.3 Local-to-global RGB-D Bundle Adjustment

To ensure that local submaps can be pieced together to form a globally consistent reconstruction of the 3D scene while maintaining real-time rates, we adopt the idea of local-to-global bundle adjustment used in [Mur-Artal et al. 2015]. After a submap $\mathcal{S}_i$ has been constructed, we select the first RGB-D scan in the submap as the submap's keyframe $\mathcal{F}_i^K$ and define the pose of $\mathcal{F}_i^K$ to be the pose of $\mathcal{S}_i$ (denoted as $\mathbf{T}_i^K$). Then we extract ORB features [Rublee et al. 2011] on the RGB channels of the keyframe $\mathcal{F}_i^K$, match them with existing features extracted from previous keyframes, and insert $\mathcal{F}_i^K$ into a global pose graph [Kümmerle et al. 2011]. The pose graph is built upon the co-visibility information between keyframes, i.e., two keyframes are connected if there are at least 15 feature matches between them. When a new keyframe $\mathcal{F}_i^K$ is added, we perform a *local bundle adjustment (local BA)* that optimizes the set of co-visible keyframes of $\mathcal{F}_i^K$ (denoted as $\mathcal{K}_{loc} = \{\mathcal{F}_k^K\}$) and all the global feature points (i.e., landmarks) $\mathcal{P}_{loc}$ observed by the keyframes in $\mathcal{K}_{loc}$. Other keyframes $\bar{\mathcal{K}}_{loc}$, which observe $\mathcal{P}_{loc}$ but are not connected to $\mathcal{F}_i^K$ in the co-visibility graph are included in the BA optimization but kept fixed. In contrast to [Mur-Artal et al. 2015], which minimizes 2D reprojection error between matched feature points, we wish to minimize the 3D geometric error directly, as we can acquire 3D feature positions from reconstructed submaps.

More specifically, the energy function to be minimized is:

$$\sum_{k=1}^{|\mathcal{K}_{loc} \cup \bar{\mathcal{K}}_{loc}|} \sum_{j=1}^{|\mathcal{M}_k|} \rho_h(\|\mathbf{p}_j^{\mathcal{S}} - \mathbf{T}_k^{K^{-1}} \mathbf{p}_j^{\mathcal{G}}\|^2_{\Omega_j^{\mathcal{S}}}), \tag{12}$$

where $\mathcal{M}_k$ is the set of matches between features in a keyframe k and landmarks in $\mathcal{P}_{loc}$, and ρ_h is the robust Huber error function [Huber 2011]. $\mathbf{p}_j^{\mathcal{S}}$ and $\Omega_j^{\mathcal{S}}$ are the 3D position and corresponding uncertainty covariance of a 2D keypoint extracted in a keyframe, respectively, which are obtained by unprojecting the 2D keypoint location onto the 3D submap and are expressed in the keyframe's local camera pose. $\mathbf{p}_j^{\mathcal{G}}$ is a 3D landmark position in $\mathcal{P}_{loc}$. Note the variables to be optimized in Eq. 12 are $\mathcal{P}_{loc}$ and the poses of the keyframes in $\mathcal{K}_{loc}$. We then use the optimized poses to update the global model.

We employ a similar strategy as in [Mur-Artal et al. 2015] to efficiently detect and correct loop closures. Loops are detected and validated using a bag of words place recognition technique, i.e., DBoW2 [Gálvez-López and Tardos 2012], and are closed by performing a pose graph optimization over a pruned version of the co-visibility graph, which retains only edges with high co-visibility and loop closure edges, to ensure robustness and efficiency. Once a loop is successfully closed, we perform an additional *global* BA to further improve the accuracy of the estimated submap poses. The global BA has a similar energy function to Eq. 12, and the main difference is that all keyframes (except the first one) and 3D landmarks need to be optimized. Both BA (local and global) and pose graph optimization are solved using the g^2o framework [Kümmerle et al. 2011] on the CPU, running simultaneously with the GPU RGB-D registration and integration thread, and thus new input RGB-D scans can be continuously registered and integrated.

Due to the BA and pose graph optimization, the pose that is used to insert a 3D submap into the global model may be different from the currently optimal one, leading to a distorted or even corrupted global model. To solve this issue, each time a BA (either local or global) or pose graph optimization is finished, we compute the delta transformation between optimized submap poses and the corresponding cached poses, then notify the GPU to transform and update the global model stored in the OpenGL texture.

5 EVALUATION AND RESULTS

We have used data streams from *a variety of consumer RGB-D cameras* to evaluate our system, including *Microsoft Kinect v2* (which includes a ToF sensor for depth sensing to achieve better data quality) and *PrimeSense* sensors (e.g. the *Asus XTion PRO LIVE* and *Structure Sensor*). Reconstruction results achieved with our system are illustrated in Figs. 1, 7–10, and 11. All times were measured on a desktop computer with an Intel Core i7 3.4GHz CPU, a NVIDIA GeForce GTX 1080 Ti graphics card, and 32GB RAM.

5.1 Uncertainty-aware Registration

As a key component of our presented system, we first separately evaluate the uncertainty-aware registration algorithm, using synthetic depth maps. The availability of ground-truth transformations between frames makes a detailed evaluation possible. We used five well-known models, which exhibit flat areas, curved surfaces, and fine details: three from the AIM@SHAPE repository (Bimba, Dancing Children, and Fandisk), the Stanford Bunny, and the Berkeley Angel [Kolluri et al. 2004]. For each model, we generated a sequence of depth maps with a ground-truth camera trajectory and added three levels of Gaussian noise to them. We set the standard deviations of the Gaussian distributions to $\sigma_n = 0.0025L$ (low noise), $\sigma_n = 0.005L$ (medium noise), and $\sigma_n = 0.01L$ (high noise), where L is the largest dimension of the scene bounding box. Then, for each model and noise level, five pairs of synthetic depth maps with different camera poses were selected. The differences in viewing angle for pairs of depth maps varied from 8° to 40°. The generated point clouds had 78,727 points on average.

We compared our uncertainty-aware registration algorithm with other ICP variants: point-to-point ICP, point-to-plane ICP, GICP [Segal et al. 2009] (using their implementations in the *Point Cloud Library* (PCL)[3] [Holz et al. 2015]), and SICP [Bouaziz et al. 2013]. In each case, the same maximum correspondence search radius and stopping criteria were used. For SICP, we used the L_p norm with $p = 0.4$ as suggested in [Bouaziz et al. 2013]. In addition, we included two weighted ICP variants: ICP (+) and ICP (++). In ICP (+), we weighted each point based on its viewing distance and viewing angle (the angle between the viewing direction and the surface normal), using the weighting function defined in [Zollhöfer et al. 2015]; while for ICP (++), each point was weighted using the confidence weight c_{nk} (Eq. 10). Both ICP (+) and ICP (++) inherit the point-to-plane distance metric. To evaluate the performance for wide baseline matching, we also compared our algorithm with several global registration methods: Super4PCS [Mellado et al. 2014], GoICP [Yang et al. 2013], and FGR [Zhou et al. 2016]. For Super4PCS, we used the *extreme* setting suggested by the authors (for high accuracy); for GoICP, we used 1,000 sample points with 10% trimming rate; the authors' parameters were used for FGR. Note in this experiment, we only minimized the geometric term E_{geo} (Eq. 7) in our algorithm, as the color information is not available.

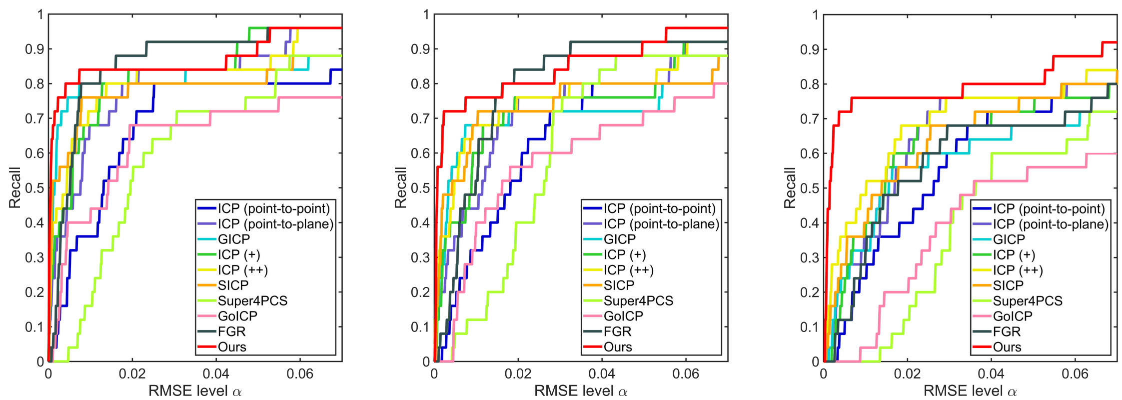
Table 1 summarizes the average and maximal RMSE for each method, and for each different noise level. Note that we compute the RMSE based on the ground-truth correspondences after alignment. Fig. 5 shows the accuracy of each algorithm for different RMSE levels. For each RMSE level α, the fraction of tests for which the algorithm achieves RMSE $< \alpha$ is plotted (higher is better). In Fig. 6, we show the changing trend of average RMSE with increasing baseline.

Our approach outperforms all other methods when a significant level of noise is presented in the input data ($\sigma_n >= 0.005L$), and at the highest noise level ($\sigma_n = 0.01L$), our algorithm is best by a big margin. Our approach is not affected by the increasing noise, since we use a noise-resilient energy function (Eq. 7), which effectively suppresses the effect of noisy matches. Point-to-plane ICP and GICP only consider the sampling uncertainty within the local tangent plane, so do not perform well on noisy point sets. ICP (+) shows only marginal improvements over point-to-plane ICP, while ICP (++) improves registration accuracy noticeably in high noise level cases. However, the *scalar* confidence weight cannot

[3] http://pointclouds.org

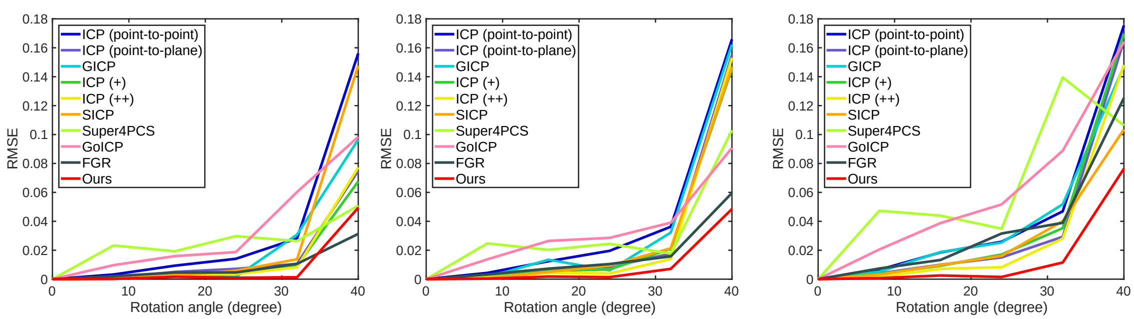
Table 1. Average and maximal RMSE for each method, for different noise levels.

	$\sigma_n = 0.0025L$		$\sigma_n = 0.005L$		$\sigma_n = 0.01L$	
	Avg. RMSE	Max. RMSE	Avg. RMSE	Max. RMSE	Avg. RMSE	Max. RMSE
ICP (point-to-point)	0.042	0.331	0.048	0.332	0.055	0.303
ICP (point-to-plane)	0.020	0.199	0.038	0.433	0.045	0.282
GICP	0.026	0.193	0.043	0.258	0.050	0.231
ICP (+)	0.017	0.188	0.036	0.427	0.046	0.292
ICP (++)	0.018	0.192	0.035	0.427	0.035	0.256
SICP	0.020	0.218	0.036	0.460	0.039	0.207
Super4PCS	0.030	0.092	0.038	0.180	0.074	0.571
GoICP	0.041	0.183	0.040	0.151	0.073	0.322
FGR	**0.011**	**0.082**	0.019	0.152	0.040	0.187
Ours	**0.011**	0.121	**0.012**	**0.131**	**0.019**	**0.127**



(a) $\sigma_n = 0.0025L$ (b) $\sigma_n = 0.005L$ (c) $\sigma_n = 0.01L$

Fig. 5. RMSE-recall curve for different methods, and different noise levels.



(a) $\sigma_n = 0.0025L$ (b) $\sigma_n = 0.005L$ (c) $\sigma_n = 0.01L$

Fig. 6. Average RMSE for different algorithms, for each noise level, with increasing baselines.

capture the anisotropic measurement uncertainty in different directions, thus, compared to our algorithm, ICP (++) fails to produce tight registration. SICP penalizes ourliers among correspondences, thus has similar effects to ICP (++). Nevertheless, it can not detect outliers as accurately as ICP (++), which limits its robustness against data noise. Instead of using dense correspondences, Super4PCS optimizes the alignment by matching tuples of points and is thus possibly more vulnerable to highly noisy input (see Figs. 5(c) and 6(c)). A similar observation holds for GoICP, where the use of full correspondences is intractable due to computational cost, and only a subset of input points can be used for registration. FGR utilizes line processes for robust alignment estimation, so performs better than Super4PCs and GoICP with respect to noise. However, when noisy correspondences dominate the correspondence set ($\sigma_n = 0.01L$), even with robust estimation, it is still difficult to produce satisfactory results. Note that in the low noise case ($\sigma_n = 0.0025L$), FGR obtains better results for average and maximal RMSE owing to its better convergence in wide baseline cases (see Fig. 6(a)).

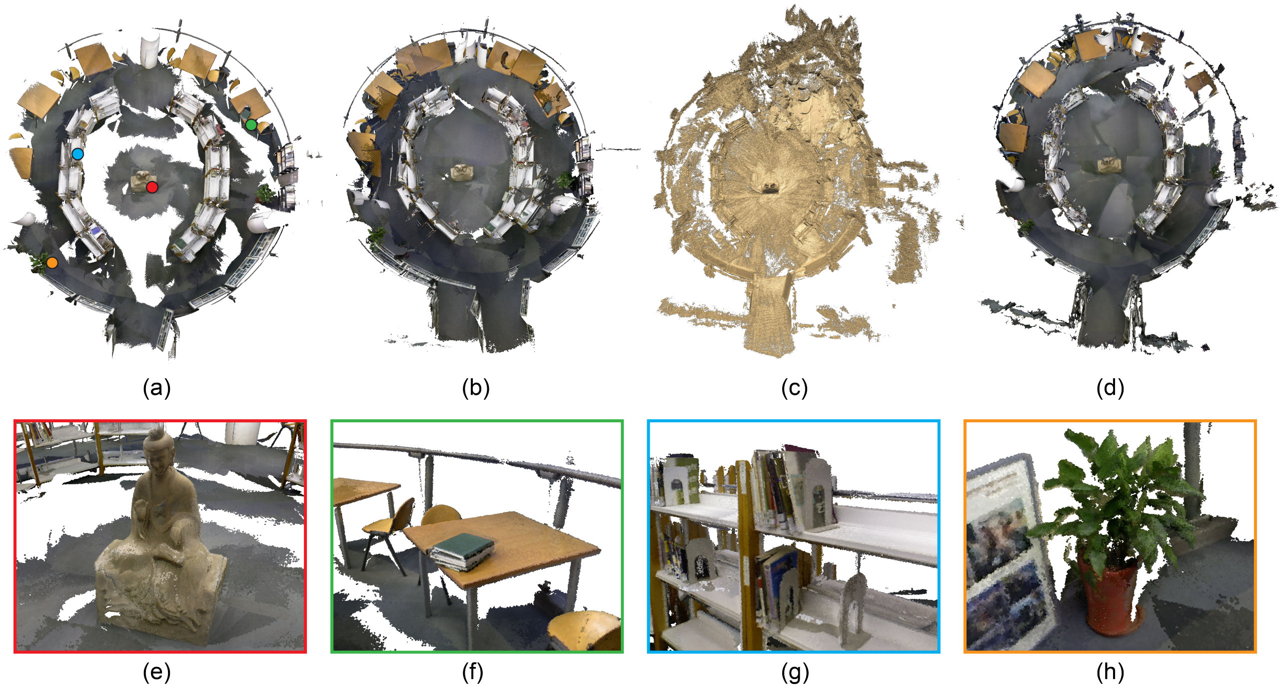


Fig. 7. Comparison of our reconstruction result of a reading area with ElasticFusion [Whelan et al. 2015b], the Redwood system [Choi et al. 2015], and BundleFusion [Dai et al. 2017]. (a): Top view of the 3D model reconstructed by our method. (b): Reconstruction result of ElasticFusion. (c): Reconstruction result of the Redwood system. Note the Redwood system uses only geometric information for global registration and outputs meshes without color. (d): Reconstruction result of BundleFusion (4 mm voxels). (e-h): Scene details reconstructed by our approach. Note our algorithm only integrates points whose measurement uncertainty is below a threshold to ensure reconstruction accuracy (see Section 3.2.2), which causes some missing data on the floor.

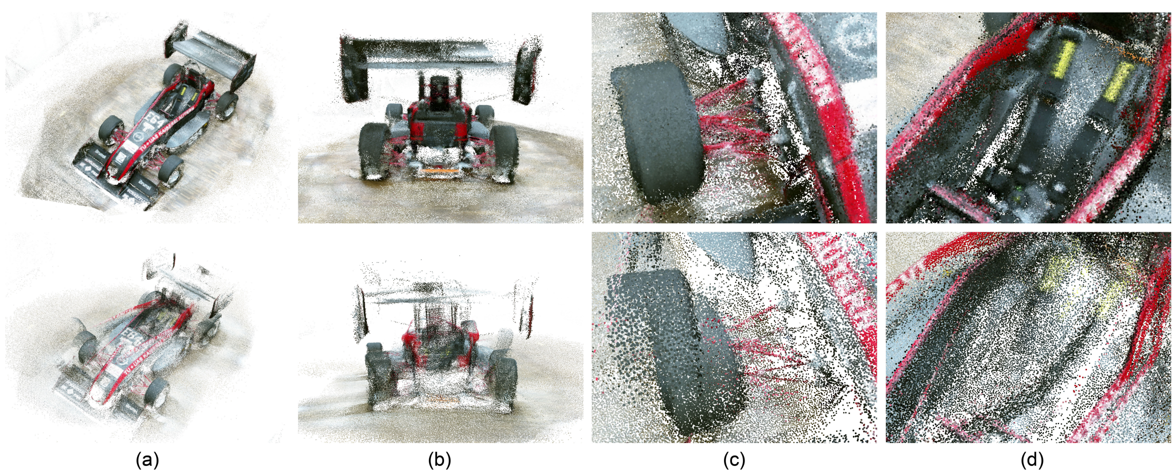


Fig. 8. Comparison of our reconstruction result (top row) of a racing car with ElasticFusion [Whelan et al. 2015b] (bottom row). (a): Front view. (b): Back view. (c): A closer inspection of the reconstructed wheel and axle. (d): A closer inspection of the reconstructed car seat.

5.2 Reconstruction

We have verified our reconstruction system using both public RGB-D datasets and RGB-D streams captured by ourselves. In Figs. 7 and 8 we compare the reconstruction results of our approach with those of ElasticFusion [Whelan et al. 2015b]; while in Figs. 7 and 9 we compare them with the Redwood system [Choi et al. 2015]. The reading area scene in Fig. 7 has many glossy bookshelves and thin structures, so the depth maps captured by a consumer RGB-D camera (we used an *Asus XTion PRO LIVE*) contain a significant amount of noise and *stray pixels* far from the surface. Furthermore, in order to capture as many scene details as possible, the camera followed a

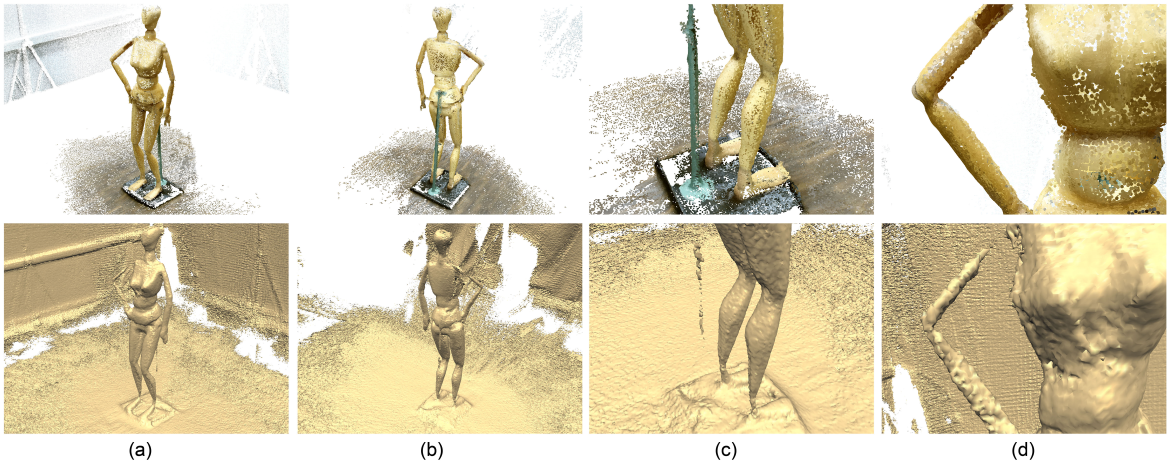


Fig. 9. Comparison of our reconstruction result (top row) of a wooden human model with the Redwood system [Choi et al. 2015] (bottom). (a): Front view. (b): Back view. (c): A closer inspection of the reconstructed legs and foothold. (d): A closer inspection of the reconstructed arm. Our approach only integrates points whose measurement uncertainty is below a threshold to ensure reconstruction accuracy (see Section 3.2.2), which causes some missing data on the walls and floor.

complex trajectory, moving up and down, and back and forth, frequently, with fast camera motions, and many small loops. In this case, ElasticFusion and Redwood[4] failed to produce satisfactory results (see Fig. 7(b,c)). ElasticFusion successfully generated a set of locally consistent segments by deforming the fused model, but it failed to close global loops due to the great amount of accumulated tracking drift. The substantial measurement noise corrupted the local fragments of the Redwood system, reducing the number of reliable pairwise alignments between fragments, and hence deteriorating the result of global fragment optimization. In contrast, our uncertainty-aware registration method proved robust to measurement noise and kept tracking drift to a minimum level; local and global BA further corrected the remaining drift between submaps.

In Figs. 8 and 9, we used two RGB-D sequences from the CoRBS benchmark [Wasenmüller et al. 2016], showing a racing car and a wooden human model. Although covering a smaller area, the fast camera motion (averaging 0.27 m/s translationally and 31.10 deg/s rotationally) still caused serious drift in ElasticFusion (see Fig. 8, bottom row). The Redwood system (with the non-rigid variant) was able to produce a consistent global model, but the measurement noise on the distant wall resulted in a noticeable amount of alignment error, distorting the fine geometric details on the human model (see Fig. 9 bottom row, (c,d)). Our approach handled the noise and reconstructed models without noticeable drift, preserving geometric details of the objects (see Figs. 8 and 9, top rows).

Figs. 7 and 10 show comparisons with BundleFusion [Dai et al. 2017]. BundleFusion globally optimizes both sparse and dense correspondences between frames in a hierarchical manner, achieving high-quality 3D reconstruction results. It uses a volumetric scene representation and so can produce smooth surfaces. However, in fig. 7(d), due to the fast camera motion and occurrence of textureless areas, BundleFusion missed *1380 out of 11250 frames* during the registration, causing noticeable missing areas and corrupted fragments around the right top corner. Furthermore, since the pose optimization process in BundleFusion does not take account of sensor noise, it produced some noisy and distorted reconstructions around the left part of the bookshelves. The same artifacts can also be observed in the insets of Fig. 10(f, g), e.g., edges of the table, chairs, and the backpack. In comparison, our approach produced noise-free and significantly higher-quality results (see Fig. 7(a, e-h) and Fig. 10 (h)).

We further evaluated the effect of each single component of our system in Fig. 10(a-e). Using the distance-and-obliqueness-weighted or the confidence-weighted ICP metric (i.e., ICP (+) or ICP (++) in Section 5.1, respectively) as the geometric term E_{geo} in Eq. 7 led to broken fragments in highly-noisy areas, e.g., around the highly reflective TV set (see Fig. 10(a,b)), although the overall global consistency of reconstructed models was guaranteed by the local-to-global RGB-D BA. On the other hand, as the meeting room has two long passages with a lot of flat surfaces and repetitive structures, using the geometric term alone (i.e., without using the photometric term E_{rgb}) led to frequent intra-submap drift which cannot be corrected by submap-level RGB-D BA (see Fig. 10(c)). In Fig. 10(d), only the pose graph optimization and global RGB-D BA were used (i.e., without the local BA); even though our system successfully detected and closed the loop, accumulated tracking error between submaps cannot be distributed and reduced to an acceptable level. Fig. 10(e) shows the reconstruction result without using the pose graph optimization and global BA, where the system was able to reconstruct locally accurate fragments but failed to close the loop and produce a geometrically consistent model.

Fig. 11 compares our reconstruction results with the mo-cap trajectory provided by the CoRBS benchmark and also illustrates the effect of our uncertainty-aware scan integration algorithm. In the top row of Fig. 11 we show the results of using the trajectory computed by our algorithm, while the bottom row shows the results of

[4]We use the rigid variant of their approach, as it performed better than the non-rigid variant in this case.

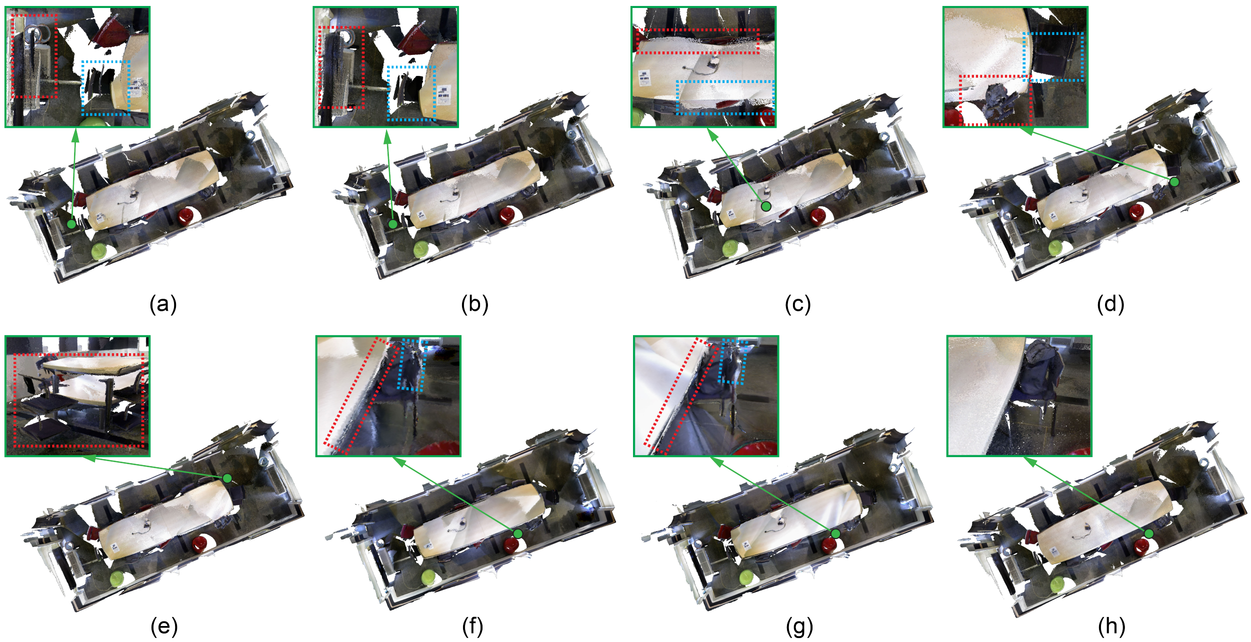


Fig. 10. Evaluation of our system and comparison with BundleFusion [Dai et al. 2017], showing a reconstructed meeting room scene (captured by an *Asus XTion PRO LIVE* sensor). (a)-(e) show results of our system, with different options, evaluating each single component of the proposed system. (a) Use ICP with distance and viewing angle based importance weighting for the geometry term E_{geo}. (b): Use confidence-weighted ICP metric for E_{geo}. (c): Without the photometric term E_{rgb}. (d): Without the local RGB-D BA. (e): Without the pose graph optimization and global RGB-D BA. (f): BundleFusion result (10 mm voxel resolution). (g): BundleFusion result (4 mm voxel resolution). (h): Our result, using full components. Boxes highlight artifacts on reconstructed 3D models.

using the mo-cap trajectory. Results of using our scan integration approach and a direct integration method, i.e., replacing Eq. 10 by a simple average of corresponding points, are shown in the left and right columns, respectively. Apart from mo-cap trajectories, the CoRBS benchmark also provides the ground truth of the scene geometry. Thus, we additionally plot the RMSE between the ground truth and reconstructed geometry as well as the corresponding error distribution in heat color. Comparing the top left and bottom right figures, it can be concluded that our approach can produce camera trajectories whose accuracy even surpasses those of optical motion capture systems (see the blurred details in the bottom right figure). Comparing the left and right columns, we conclude that our scan integration algorithm can effectively reduce noise while preserving fine geometric details (see the noise around the top of the lids in the right column and fine details highlighted in dashed boxes).

Quantitative Results. We next quantitatively evaluate the trajectory and reconstruction accuracy of our approach on the ICL-NUIM RGB-D benchmark [Handa et al. 2014], which provides ground-truth camera trajectories for eight scan sequences of two synthetic indoor environments. The ground-truth 3D model of the *Living Room* is also provided for surface reconstruction quality evaluation. We evaluated our system on four trajectories for the *Living Room* scene (*lr kr0 – lr kr3*, with synthetic noise) and compared the accuracy of results with those from a number of state-of-the-art SLAM and indoor reconstruction systems: DVO SLAM [Kerl et al. 2013], RGB-D SLAM [Endres et al. 2012], MRSMAP [Stückler and Behnke 2014], Kintinouous [Whelan et al. 2015a], ElasticFusion [Whelan et al. 2015b], BundleFusion [Dai et al. 2017], [Kähler et al. 2016] and the Redwood reconstruction system [Choi et al. 2015]. Note that [Kähler et al. 2016] does not use RGB information, and Redwood runs offline. In addition, as in Figs. 10 and 11, we used this synthetic benchmark to evaluate each single component of our system, including the corresponding results for comparison as well. The trajectory accuracy is measured by absolute trajectory error (ATE), while the reconstruction quality is measured by the

Table 2. ATE RMSE on the ICL-NUIM benchmark (measured in meters).

	lr kt0	lr kt1	lr kt2	lr kt3
DVO SLAM	0.104	0.029	0.191	0.152
RGB-D SLAM	0.026	0.008	0.018	0.433
MRSMap	0.204	0.228	0.189	1.090
Kintinuous	0.072	0.005	0.010	0.355
ElasticFusion	0.009	0.009	0.014	0.106
BundleFusion	0.006	**0.004**	0.006	0.011
Redwood	0.256	0.030	0.033	0.061
Ours (distance&obliqueness-weighted)	0.019	0.016	0.037	0.043
Ours (confidence-weighted)	0.013	0.011	0.018	0.024
Ours (no photometric)	0.043	0.010	0.020	0.031
Ours (no local BA)	0.008	0.009	0.009	0.011
Ours (no global BA)	**0.005**	**0.004**	**0.005**	0.016
Ours	**0.005**	**0.004**	**0.005**	**0.010**

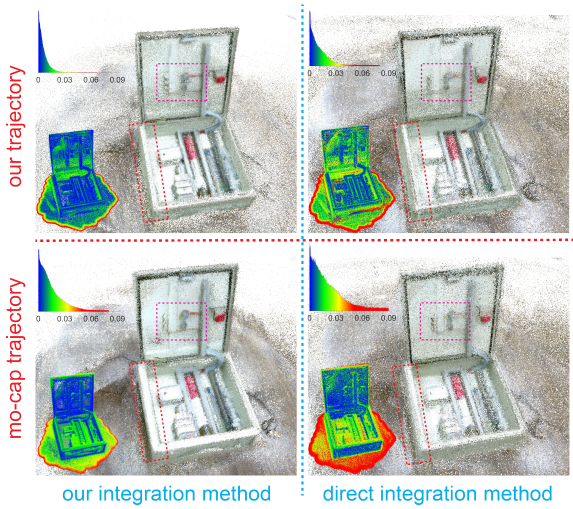


Fig. 11. Comparison of reconstruction results with mo-cap trajectory. Top left: Reconstructed model using our algorithm. Top right: Reconstruction result using our trajectory, but with a direct scan integration method. Bottom left: Reconstruction result using mo-cap trajectory, with our scan integration approach. Bottom right: Reconstruction result using mo-cap trajectory, with direct scan integration method. Boxes highlight differences between the results. Bottom insets in each image plot the reconstruction error (in heat color), and top insets show the histograms of the corresponding reconstruction error.

Table 3. Surface reconstruction error on the ICL-NUIM benchmark (measured in meters).

	lr kt0	lr kt1	lr kt2	lr kt3
DVO SLAM	0.032	0.061	0.119	0.053
RGB-D SLAM	0.044	0.032	0.031	0.167
MRSMap	0.061	0.140	0.098	0.248
Kintinuous	0.011	0.008	0.009	0.150
ElasticFusion	0.007	0.007	0.008	0.028
BundleFusion	0.005	0.006	0.007	0.008
Kähler et al.	0.013	0.011	**0.001**	0.014
Redwood	0.020	0.020	0.013	0.022
Ours (direct integration)	0.006	0.008	0.007	0.010
Ours	**0.004**	**0.005**	0.004	**0.006**

mean distance to the ground-truth surface. Tables 2 and 3 summarize the results. Our approach outperforms current state-of-the-art systems with respect to both tracking and reconstruction accuracy. Note that since these sequences were recorded with relatively simple camera trajectories, tracking accuracy of our system was not significantly affected when the local BA or global BA component was disabled (see Table 2). Furthermore, there is no loop closure in sequences *lr kr0* – *lr kr2*, so no pose graph optimization and global BA was triggered in our system. Comparing the last two rows in Table 3, our uncertainty-aware scan integration algorithm significantly improves the geometry quality.

Besides, we also evaluated the trajectory estimation accuracy of our approach on the RGB-D benchmark of [Sturm et al. 2012], which provides mo-capped ground-truth camera poses for sequences captured by a *Microsoft Kinect v1*. We selected four widely used sequences, i.e., *fr1/desk*, *fr2/xyz*, *fr3/office* and *fr3/nst*, and compared the ATE RMSE of our approach with the same set of state-of-the-art RGB-D reconstruction methods as in Table 2. Table 4 summarizes the results. For these simple benchmark sequences, our algorithm outperformed most of the state-of-the-art methods in terms of tracking quality. Note the *fr3/nst* sequence only covers a flat wall, and thus the tracking result did not benefit much from our noise-tolerant scan registration algorithm. Only *fr3/office* contains a global loop.

Table 4. ATE RMSE on the TUM RGB-D benchmark (measured in meters).

	fr1/desk	fr2/xyz	fr3/office	fr3/nst
DVO SLAM	0.021	0.018	0.035	0.018
RGB-D 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
Ours (distance&obliqueness-weighted)	0.029	0.016	0.031	0.016
Ours (confidence-weighted)	0.023	0.012	0.024	0.014
Ours (no photometric)	0.017	0.007	0.015	0.449
Ours (no local BA)	0.033	0.009	0.025	0.093
Ours (no global BA)	**0.015**	**0.006**	0.037	0.014
Ours	**0.015**	**0.006**	**0.009**	0.014

Performance. We plot the average surfel number and components of *GPU* frame processing time of our system across the *Meeting Room* sequence (see Fig. 10) in Fig. 12(a). Here, initialization includes the uncertainty model estimation and depth map filtering (Section 3.1). With an overall average 23 ms (~ 43 Hz) and maximal 29 ms (~ 34 Hz) frame processing time, our reconstruction system is capable of capturing 3D scenes in real-time. Among all the components, the uncertainty-aware registration takes around half of the frame processing time, i.e., 11 ms on average; while initialization and scan integration take 7 ms and 5 ms on average, respectively. Additionally, Table 5 summarizes the average and maximal GPU frame processing time for each sequence used in the experiments and shows that our reconstruction system achieves stable performance across a variety of 3D scenes. Note the BA (local and global) and pose graph optimization (Section 4.3) runs separately in different CPU threads, thus we can construct submaps (i.e., tracking and integrating input frames) continuously on the GPU. Furthermore, in our experiments, each time a BA is finished, the global model can be updated on the GPU within 5 ms. This adds only a slight overhead to the GPU processing time and does not frequently happen during scanning, as we only perform BA when a new submap is constructed, or a loop closure is detected. More analysis on CPU running times can be found in the supplementary material. As a comparison, Fig. 12(b,c) plot the frame processing time of BundleFusion, using a single GPU and two GPUs[5], respectively. When only a single GPU is available (i.e., Fig. 12(b)), the running time of BundleFusion grows approximately linearly as the frame number increases; this is mainly caused by the linear-growth of the scale of the global pose optimization problem as well as the relatively large amount of landmark constraints, which, under the single GPU setting, has to be solved sequentially with other components, making the frame processing time exceed the real-time limit after only a few hundred frames. Note our system runs even faster than the dual-GPU version of BundleFusion (see Fig. 12(a,c) for two reasons: (1) our system requires no time-consuming re-integration operations; (2) we produce less submaps and optimize only *sparse* correspondences among *a small set of* co-visible submaps on-the-fly.

Table 5. Average and maximal frame processing time for each sequence (measured in milliseconds).

Sequence	# Frames	Avg. time	Max. time
Lounge room (Fig. 1)	8047	22	28
Reading area (Fig. 7)	11250	26	35
Meeting room (Fig. 10)	5772	23	29
CoRBS R2 (Fig. 8)	3209	24	37
CoRBS H1 (Fig. 9)	1468	20	27
CoRBS E2 (Fig. 11)	1902	22	29

6 CONCLUSION AND DISCUSSION

We have presented an integrated approach for high-accuracy 3D reconstruction using data from consumer-grade RGB-D cameras. Our approach takes account of the inherent sensor measurement noise using an uncertainty model for each measured point while keeping the overall registration procedure under the ICP framework, and significantly outperforms previous approaches when dealing with noisy input scans. The presented pointwise uncertainty model also guides the integration of depth points in a smarter manner, reducing noise while preserving geometric details. Furthermore, we also utilize a submap-based, uncertainty-aware and local-to-global RGB-D bundle adjustment strategy to deliver a globally consistent model. We have implemented the presented algorithm on the GPU, building up a real-time 3D reconstruction system which is ready for practical use.

Limitations. One limitation of our algorithm is that although the quality of the submaps is guarded by our noise-tolerant registration algorithm, in some cases, tracking drift still accumulates to a non-negligible level. Such intra-submap drift cannot be corrected by downstream steps in our pipeline, leading to artifacts in the final reconstruction results (see Fig. 13). Ideally, we could avoid intra-submap drift by using smaller submaps and applying the submap-level bundle adjustment to a larger set of submaps. While this would increase the computational cost and prevent real-time processing rates, it could be used as an offline post-process to obtain maximum quality.

[5]We used an additional NVIDIA GeForce GTX 1080 Ti in this experiment.

Efficient data structures for volumetric fusion have been proposed [Kähler et al. 2015; Nießner et al. 2013] which store only sparse blocks of the TSDF around the actual surfaces, significantly improving the available spatial resolution of the volumetric representation. Applying our uncertainty model to such a volumetric scene representation is an interesting problem which we leave as a future work.

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

S. Bouaziz, A. Tagliasacchi, and M. Pauly. 2013. Sparse Iterative Closest Point. *Computer Graphics Forum (Symposium on Geometry Processing)* 32, 5 (2013), 1–11.

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. Y. 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

Y. Cui, S. Schuon, S. Thrun, D. Stricker, and C. Theobalt. 2013. Algorithms for 3d shape scanning with a depth camera. *IEEE transactions on pattern analysis and machine intelligence* 35, 5 (2013), 1039–1050.

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).* ACM, New York, NY, USA, 303–312. https://doi.org/10.1145/237170.237269

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

J. R. Diebel, S. Thrun, and M. Brünig. 2006. A Bayesian Method for Probable Surface Reconstruction and Decimation. *ACM Trans. Graph.* 25, 1 (Jan. 2006), 39–59. https://doi.org/10.1145/1122501.1122504

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.* 1691–1696. https://doi.org/10.1109/ICRA.2012.6225199

F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard. 2014. 3-D Mapping With an RGB-D Camera. *IEEE Transactions on Robotics* 30, 1 (Feb 2014), 177–187. https://doi.org/10.1109/TRO.2013.2279412

J. Engel, T. Schöps, and D. Cremers. 2014. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 834–849. https://doi.org/10.1007/978-3-319-10605-2_54

D. Ferstl, C. Reinbacher, G. Riegler, M. Rüther, and H. Bischof. 2015. Learning Depth Calibration of Time-of-Flight Cameras. In *Proceedings of the British Machine Vision Conference (BMVC)*, Mark W. Jones Xianghua Xie and Gary K. L. Tam (Eds.). BMVA Press, Article 102, 12 pages. https://doi.org/10.5244/C.29.102

N. Fioraio, J. Taylor, A. Fitzgibbon, L. Di Stefano, and S. Izadi. 2015. Large-scale and drift-free surface reconstruction using online subvolume registration. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 4475–4483. https://doi.org/10.1109/CVPR.2015.7299077

S. Fuhrmann and M. Goesele. 2014. Floating Scale Surface Reconstruction. *ACM Trans. Graph.* 33, 4, Article 46 (July 2014), 11 pages. https://doi.org/10.1145/2601097.2601163

D. Gálvez-López and J. D. Tardos. 2012. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics* 28, 5 (Oct 2012), 1188–1197. https://doi.org/10.1109/TRO.2012.2197158

B. Glocker, J. Shotton, A. Criminisi, and S. Izadi. 2015. Real-Time RGB-D Camera Relocalization via Randomized Ferns for Keyframe Encoding. *IEEE Transactions on Visualization and Computer Graphics* 21, 5 (May 2015), 571–583. https://doi.org/10.1109/TVCG.2014.2360403

G. Grigoryan and P. Rheingans. 2004. Point-based probabilistic surfaces to show surface uncertainty. *IEEE Transactions on Visualization and Computer Graphics* 10, 5 (Sept 2004), 564–573. https://doi.org/10.1109/TVCG.2004.30

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.

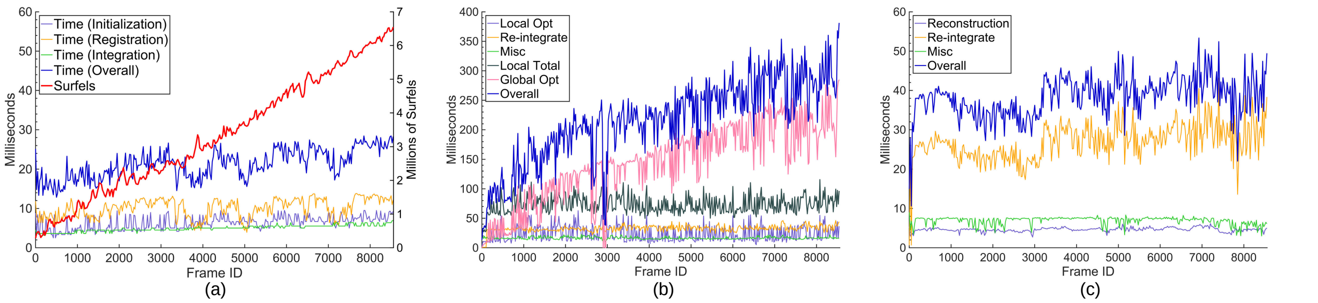


Fig. 12. Details of frame processing time. (a): Our system (the right axis for surfel numbers). (b): BundleFusion system, single GPU. (c): BundleFusion system, two GPUs (only the timing on the main GPU is plotted, as two GPUs runs in parallel.

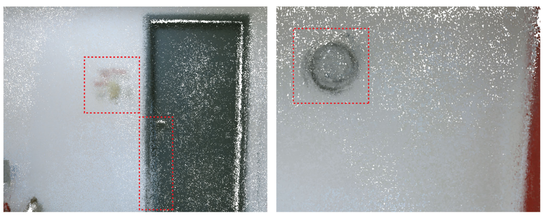


Fig. 13. Limitation. Drifts within submaps cannot be corrected by submap-level optimization, thus cause visible artifacts in the reconstructed models. Areas with artifacts are highlighted with red boxes.

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. *Int. J. Rob. Res.* 31, 5 (April 2012), 647–663. https://doi.org/10.1177/0278364911434148

D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke. 2015. Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D. *IEEE Robotics Automation Magazine* 22, 4 (Dec 2015), 110–124. https://doi.org/10.1109/MRA.2015.2432331

P. J. Huber. 2011. *Robust statistics*. Springer.

P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Strasser. 2006. Bayesian Point Cloud Reconstruction. *Computer Graphics Forum* (2006). https://doi.org/10.1111/j.1467-8659.2006.00957.x

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 (Aug 2011), 1633–1645. https://doi.org/10.1109/TPAMI.2010.223

A. Jordt and R. Koch. 2013. *Reconstruction of Deformation from Depth and Color Video with Explicit Noise Models*. Springer Berlin Heidelberg, Berlin, Heidelberg, 128–146. https://doi.org/10.1007/978-3-642-44964-2_7

O. Kähler, V. A. Prisacariu, and D. W. Murray. 2016. *Real-Time Large-Scale Dense 3D Reconstruction with Loop Closure*. Springer International Publishing, Cham, 500–516. https://doi.org/10.1007/978-3-319-46484-8_30

O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray. 2015. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Devices. *IEEE Transactions on Visualization and Computer Graphics* 21, 11 (Nov 2015), 1241–1250. https://doi.org/10.1109/TVCG.2015.2459891

A. Kalaiah and A. Varshney. 2003. Statistical Point Geometry. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (SGP '03)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 107–115. http://dl.acm.org/citation.cfm?id=882370.882385

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 *2013 International Conference on 3D Vision - 3DV 2013*. 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

Y. M. Kim, D. Chan, C. Theobalt, and S. Thrun. 2008. Design and calibration of a multi-view TOF sensor fusion system. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 1–7. https://doi.org/10.1109/CVPRW.2008.4563160

R. Kolluri, J. Richard Shewchuk, and J. F. O'Brien. 2004. Spectral Surface Reconstruction from Noisy Point Clouds. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (SGP '04)*. ACM, New York, NY, USA, 11–21. https://doi.org/10.1145/1057432.1057434

R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. 2011. G2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*. 3607–3613. https://doi.org/10.1109/ICRA.2011.5979949

F. Lenzen, K. I. Kim, H. Schäfer, R. Nair, S. Meister, F. Becker, C. S. Garbe, and C. Theobalt. 2013. *Denoising Strategies for Time-of-Flight Data*. Springer Berlin Heidelberg, Berlin, Heidelberg, 25–45. https://doi.org/10.1007/978-3-642-44964-2_2

H. Li, E. Vouga, A. Gudym, L. Luo, J. T. 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

R. Maier, J. Sturm, and D. Cremers. 2014. Submap-based bundle adjustment for 3D reconstruction from RGB-D data. In *German Conference on Pattern Recognition*. Springer, 54–65.

N. Mellado, D. Aiger, and N. J. Mitra. 2014. Super 4PCS Fast Global Pointcloud Registration via Smart Indexing. *Computer Graphics Forum* 33, 5 (2014), 205–215. https://doi.org/10.1111/cgf.12446

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 (Oct 2015), 1147–1163. https://doi.org/10.1109/TRO.2015.2463671

A. Myronenko and X. Song. 2010. Point Set Registration: Coherent Point Drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 12 (Dec 2010), 2262–2275. https://doi.org/10.1109/TPAMI.2010.46

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

C. V. Nguyen, S. Izadi, and D. Lovell. 2012. Modeling Kinect Sensor Noise for Improved 3D Reconstruction and Tracking. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*. 524–530. https://doi.org/10.1109/3DIMPVT.2012.84

M. Nießner, M. Zollhöfer, 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

E. Olson and P. Agarwal. 2013. Inference on Networks of Mixtures for Robust Robot Mapping. *Int. J. Rob. Res.* 32, 7 (June 2013), 826–840. https://doi.org/10.1177/0278364913479413

J. H. Park, Y. D. Shin, J. H Bae, and M. H. Baeg. 2012. Spatial uncertainty model for visual features using a Kinect sensor. *Sensors* 12, 7 (2012), 8640–8662.

R. R. Paulsen, J. A. Baerentzen, and R. Larsen. 2010. Markov Random Field Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics* 16, 4 (July 2010), 636–646. https://doi.org/10.1109/TVCG.2009.208

F. Reichl, J. Weiss, and R. Westermann. 2016. Memory-Efficient Interactive Online Reconstruction From Depth Image Streams. In *Computer Graphics Forum*. Wiley Online Library.

M. Reynolds, J. Doboš, L. Peel, T. Weyrich, and G. J. Brostow. 2011. Capturing Time-of-Flight data with confidence. In *CVPR 2011*. 945–952. https://doi.org/10.1109/CVPR.2011.5995550

H. Roth and M. Vona. 2012. Moving Volume KinectFusion. In *British Machine Vision Conference*. British Machine Vision Association, 112.1–112.11.

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*. 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544

M. Ruhnke, R. Kümmerle, G. Grisetti, and W. Burgard. 2012. Highly accurate 3D surface models by sparse surface adjustment. In *2012 IEEE International Conference on Robotics and Automation*. 751–757. https://doi.org/10.1109/ICRA.2012.6225077

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

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

F. Steinbrücker, C. Kerl, and D. Cremers. 2013. Large-Scale Multi-resolution Surface Reconstruction from RGB-D Sequences. In *2013 IEEE International Conference on Computer Vision*. 3264–3271. https://doi.org/10.1109/ICCV.2013.405

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.

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*. 573–580. https://doi.org/10.1109/IROS.2012.6385773

A. Teichman, S. Miller, and S. Thrun. 2013. Unsupervised Intrinsic Calibration of Depth Sensors via SLAM.. In *Robotics: Science and Systems*, Vol. 248.

C. Wang and X. Guo. 2017. Feature-based RGB-D camera pose optimization for real-time 3D reconstruction. *Computational Visual Media* 3, 2 (2017), 95–106.

H. Wang, J. Wang, and L. Wang. 2016. Online Reconstruction of Indoor Scenes from RGB-D Streams. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3271–3279. https://doi.org/10.1109/CVPR.2016.356

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

T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. Mcdonald. 2015a. Real-time Large-scale Dense RGB-D SLAM with Volumetric Fusion. *Int. J. Rob. Res.* 34, 4-5 (April 2015), 598–626. https://doi.org/10.1177/0278364914551008

T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. 2015b. ElasticFusion: Dense SLAM without a pose graph. *Robotics: Science and Systems* (2015).

H. Yamazoe, H. Habe, I. Mitsugami, and Y. Yagi. 2018. Depth error correction for projector-camera based consumer depth cameras. *Computational Visual Media* 4, 2 (2018), 1–9.

J. Yang, H. Li, and Y. Jia. 2013. Go-ICP: Solving 3D Registration Efficiently and Globally Optimally. In *2013 IEEE International Conference on Computer Vision*. 1457–1464. https://doi.org/10.1109/ICCV.2013.184

A. Zeng, S. Song, M. Nießner, M. Fisher, and J. Xiao. 2016. 3DMatch: Learning the Matching of Local 3D Geometry in Range Scans. *arXiv preprint arXiv:1603.08182* (2016).

M. Zeng, F. Zhao, J. Zheng, and X. Liu. 2013. Octree-based Fusion for Realtime 3D Reconstruction. *Graph. Models* 75, 3 (May 2013), 126–136. https://doi.org/10.1016/j.gmod.2012.09.002

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

Q. Y. Zhou, J. Park, and V. Koltun. 2016. *Fast Global Registration*. Springer International Publishing, Cham, 766–782. https://doi.org/10.1007/978-3-319-46475-6_47

M. Zollhöfer, A. Dai, M. Innmann, C. Wu, M. Stamminger, C. Theobalt, and M. Nießner. 2015. Shading-based Refinement on Volumetric Signed Distance Functions. *ACM Trans. Graph.* 34, 4, Article 96 (July 2015), 14 pages. https://doi.org/10.1145/2766887