Computer-Aided Design 102 (2018) 33–43

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Sparse3D: A new global model for matching sparse RGB-D dataset with small inter-frame overlap

Canyu Le [a], Xin Li [b,*]

[a] Xiamen University, China
[b] Louisiana State University, United States

ARTICLE INFO

Keywords:
Sparse SLAM
Sparse RGB-D reconstruction
Partial matching

ABSTRACT

We present a novel 3D global matching algorithm, *Sparse3D*, to handle the challenging reconstruction of RGB-D datasets whose inter-frame overlap is small due to insufficient temporal sampling or fast camera movement. To support a more reliable reconstruction, two major technical components are proposed: (1) pairwise alignment using a set of complementary features, and (2) a novel global model for alignment pruning and pose optimization. We examine the effectiveness of our algorithm on multiple benchmark datasets under various inter-frame overlap, and demonstrate it better reliability over existing RGB-D reconstruction algorithms.

1. Introduction

3D scene reconstruction from RGB-D datasets is a fundamental problem in geometric modeling, and it has many applications in computer graphics, vision, and robotics. Reconstruction algorithms can be generally categorized into two types. One is the *real-time* reconstruction, which is often studied in visual SLAM that simultaneously localizes the camera (robot) and maps its surrounding environment. To direct the movement of robot, the reconstruction and localization often need to be done instantaneously. The second type is the *offline* reconstruction, which processes the images/scans after their acquisition. Without the restriction on real-time processing, more sophisticated matching and analysis schemes can be adopted in an offline algorithm to handle more challenging datasets such as highly noisy or unorganized (e.g., non-sequential) input.

Most existing frame registration and reconstruction algorithms, especially the real-time approaches, rely on the assumption that consecutive frames exhibit significant overlap. For example, many of these algorithms adopt the ICP or its variant strategies, whose alignment could easily get trapped in local optima when initial guesses are not good. Therefore, when using these algorithms, if the neighboring frames have significant rotations or shifts, which could be due to fast camera movements or low scanning frame rates, the frame matching and camera pose tracking often become unreliable. Fig. 1(a) and (b) illustrate the conventionally considered reconstruction scenarios (a), and this more general but challenging scenario (b), respectively.

Our goal is to develop reconstruction algorithms that can effectively match and stitch frames with small overlap, which existing algorithms often failed (e.g. Fig. 1c). It could make RGB-D reconstructions applicable to those scans that are sequentially obtained but have small inter-frame redundancy, or even to the un-ordered datasets (Fig. 1d). Our idea is based on the following observations.

(1) ICP and its variants, used in popular SLAM/reconstruction frameworks such as *Kintinuous* [1] and *ElasticFusion* [2] and [3], are sensitive to initial poses. To fundamentally overcome this limitation when dense sampling frame rate and big overlap are not guaranteed, feature guidance is necessary in frame matching.
(2) Matching guided by single features, used in recent real-time algorithms such as *ORB-SLAM* [4], *ORB-SLAM2* [5] and *SfM* (Structure-from-Motion) [6,7], cannot guarantee the correct pairwise alignment when overlap is small. Features are often designed to reflect certain invariance and need to find a balance between being discriminative and being too sensitive, hence, different features work better under different scenarios. To improve the feature reliability under various scenarios, integrating a set of complementary features could be effective.
(3) Local pairwise alignments are sometimes inevitably ambiguous (see Fig. 5 for an example). Preserving multiple potential alignments and then pruning them through a global aspect is often more robust.

Based on these observations, we first propose to compute pairwise matching using multiple features which complementarily

* Corresponding author.
E-mail addresses: lecanyu@gmail.com (C. Le), xinli@lsu.edu (X. Li).

https://doi.org/10.1016/j.cad.2018.04.018
0010-4485/

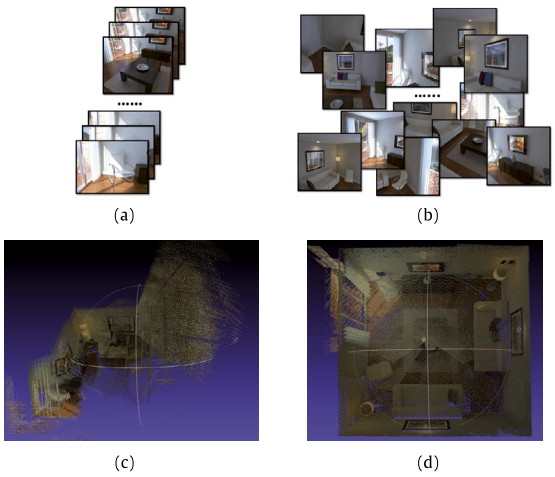


Fig. 1. Reconstructions from (a) sequential datasets with big inter-frame overlap versus from (b) unordered datasets with small inter-frame overlap. Most existing reconstruction algorithms do not work well when dealing with scenario (b). (c) The reconstruction result from (b) (but the data are ordered in sequence) using *Kintinuous* [1] where obvious drifts and errors exist; (d) our reconstruction result from (b). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

capture geometry and texture properties, and keep multiple potential alignments. This could make the computed feature correspondence more reliable. Then, since pairwise alignment could still be error-prone or ambiguous, we do a global pruning that analyzes the mutual consistency among multiple related frames, to effectively improve the overall reconstruction robustness. The two technical contributions are summarized as follows.

- To improve the reliability of pairwise alignment, we propose to use multiple complementary features, to get multiple potential correspondences (alignments) between each frame pair. This strategy could improve the probability of finding correct feature correspondence.
- We propose a new global pruning and optimization model to maximize mutual consistency of alignments among multiple frames. It can effectively identify correct pairwise alignments even when the number of outliers is big. When dealing with the challenging sparsely sampled datasets, this model could greatly improve the reconstruction success rate.

2. Related work

2.1. Real-time reconstruction systems

Many recent SLAM systems are developed to perform reconstructions in real-time. A classic RGB-D SLAM system is *KinectFusion* [8] that runs on RGB-D video streams. It uses ICP algorithm to track camera pose. A major limitation is its incapability in processing long video stream and big environment, due to the lack of effective elimination of accumulative errors. Subsequently, the *Kintinuous* [1] was proposed to better tackle this. It utilizes place recognition [9] to detect loop and performs a pose graph optimization [10] to suppress accumulative error. More recently, *ElasticFusion* [2] adopts a *surfel-based map* of the environment, rather than the pose-graph optimization to reduce accumulative error. Both *Kintinuous* and *ElasticFusion*, however, only work well on dense data stream (in which inter-frame overlap is big) because of the sensitivity of their ICP-based registration strategy.

Another recent SLAM system, *ORB-SLAM2* [5], integrates many advanced techniques like *ORB* feature detection [11], place recognition [9], and bundle adjustment [12] in the reconstruction pipeline. Despite its great robustness dealing with dense input, it still fails when processing data with small inter-frame overlap. Because the *ORB* [11] itself still cannot provide robust enough corresponded features in various scenarios when overlap is small, and could fail to align some frames. Also, when feature outliers significantly increase, the tracking module often becomes 20–30 times slower in finding the matching and prone to failure, and hence, becomes unstable (two executions may produce different results).

Another state-of-the-art system, the *BundleFusion* [13], combines three main strategies: sparse-to-dense alignment, local-to-global optimization, and dynamic *TSDF* in their reconstruction. But its performance on sparse datasets are limited: (1) Locally, SIFT feature are sometimes unreliable; and local ambiguity (i.e. locally well aligned but globally incorrect. Also see Fig. 5) cannot be handled even with the validation mechanism of [13]. (2) Globally, the hierarchical optimization is efficient. But if some local pairwise alignments are incorrect, the optimization will be affected and could result in incorrect composition.

2.2. Off-line reconstruction systems

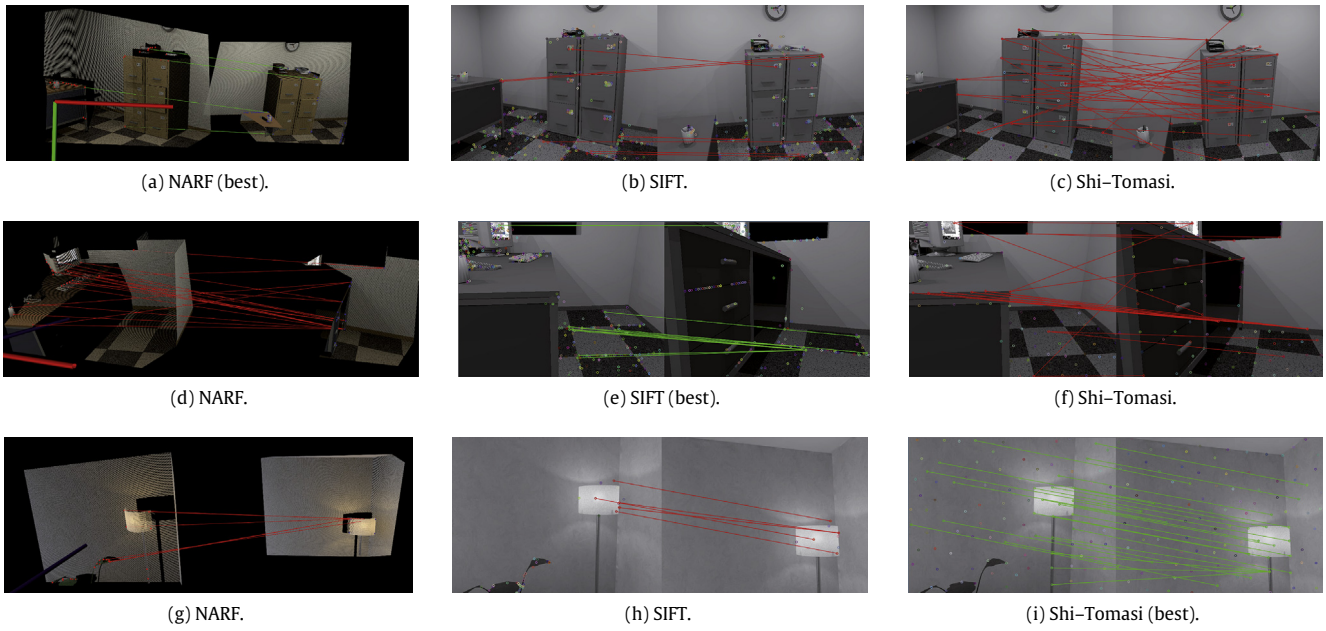
The earlier and famous offline reconstruction system is the *VisualSFM* [6] which is based on *Structure from Motion (SfM)* technique. This system could reconstruct from multiple unordered RGB images by simulating multi-view model. Recently, an augmented *SfM*, called *COLMAP* [7], was developed to perform large-scale reconstruction from unordered Internet photo collections. While it adopted multiple strategies to enhance matching robustness and accuracy, the feature extraction and correspondence calculation are still sometimes unreliable in processing frames with small overlap. From these alignments, its subsequent greedy *next-best-view-selection* still cannot find good matching to produce correct reconstruction.

Choi et al. [3] integrate several consecutive frames into a big fragment using an ICP frame-to-model registration, then apply a global optimization based on [14]. This system could detect and delete some false loops and achieve relative high accuracy in large 3D environment. But when processing scans with small inter-frame overlap, it is still not robust enough, because (1) ICP-based stitching is sensitive to bad local optima in handling big inter-frame shift, and (2) the global optimization is sensitive to initial guess which often does not perform well when inter-frame overlap is small.

Zheng et al. [15] proposed a multi-frame graph matching algorithm to simultaneously match several consecutive frames. A big affinity matrix is used to encode spatial coherency of features from multiple frames, to enhance robustness of feature correspondence. Wang et al. [16] use SURF features and simultaneously match multiple frames using a feature correspondence list to improve the matching reliability. While these strategies demonstrate better reliability in feature matching than standard pairwise matching, when inter-frame overlap is small, having multiple frames covering a same big region is usually not easy. Hence, their performance improvement are limited.

Lin et al. [17] combine the texture and geometric features to prune and improve correspondence. The idea is to seek for a good compromise between texture consistency and geometric consistency to improve the feature correspondences.

To the best of our knowledge, most existing real-time and offline reconstruction algorithms are designed to handle the scans whose camera shift is relatively small, with correlated (consecutive) frames having significant overlap. They cannot effectively handle data with small overlap, as we will demonstrate in Section 6.



(a) NARF (best). (b) SIFT. (c) Shi–Tomasi.

(d) NARF. (e) SIFT (best). (f) Shi–Tomasi.

(g) NARF. (h) SIFT. (i) Shi–Tomasi (best).

Fig. 2. Features extracted by different detectors. These nine images show complementary behaviors of the NARF (a, d, g), SIFT (b, e, h) and Shi–Tomasi (c, f, i) detectors. The three rows show the different scenes in which these detectors behave differently: NARF performs the best in the first scene (row); SIFT performs better when dealing with occlusion and low overlap frames such as in the second scene, while Shi–Tomasi performs the best in the third scene with little texture and geometry saliency. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

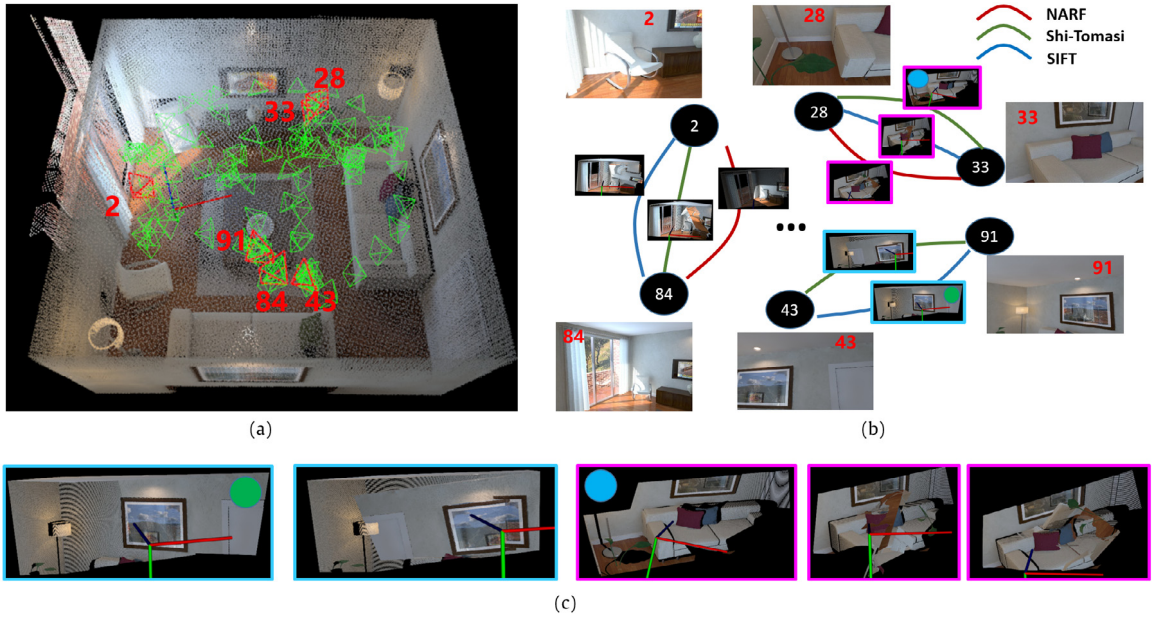


Fig. 3. 3D Reconstruction from only 115 frames of the ICL/room dataset (original 2870 frames). (a) Reconstructed living room model, together with the camera pose distribution. (b) Stitching between a few frames. Between each pair of frames (nodes), there are multiple alignments (edges) calculated using correspondences from different types of features. Some alignments are correct while some other alignments are not, as shown in the zoom-in figures (c).

3. Overview

Our reconstruction pipeline consists of three main steps: (1) feature extraction and correspondence, (2) pairwise frame matching, (3) multi-graph global optimization for alignment pruning and refinement.

Multi-Feature Detection and Matching. Desirable features in RGB-D reconstruction should have high repeatability, be invariant to viewpoint, and be insensitive to noise [18]. We aim to integrate color and geometry information into features to capture different and complementary characteristics of the given data. First, we adopt NARF [19] as a geometric detector, which was reported to be efficient and reliable in describing range images in RGB-D reconstructions [15]. But depth scans from low-cost consumer depth sensors often have very low resolution and is very noisy and unstable especially near the silhouette or occlusion boundaries. Integrating color information could help remedy these issues. We found the usage of the SIFT [20] and Shi–Tomasi [21] detectors is effective. SIFT has strong rotation, translation and scale invariance so it has high repeatability during the change of viewpoints. Although SIFT performs very well in texture-rich areas, it has

bad performance in textureless areas, as shown in Fig. 2(h). Shi–Tomasi corner detector [21], as a complement for SIFT, can extract more salient points in low texture areas, as the comparison shown in Fig. 2(h) and (i). Using complementary features can provide more alignment candidates under different scenarios and allow us to enhance the matching robustness in challenging small inter-frame overlap scenarios. Practical reconstruction examples benefited from this will be given and elaborated later (e.g. Figs. 3 and 4).

Graph Matching for Feature Correspondence. To compute the pairwise transformation between two frames, we should find one-to-one correspondence among features. RANSAC [22] is a commonly used approach. However, it could be slow and unstable when the ratio of outliers is big which is very common in small inter-frame overlap scenario. So we use the graph matching [23], a more stable algorithm, to solve the correspondence. To perform efficient graph matching, we first build a set of initial correspondences using nearest neighbors in the feature descriptor vector space; then use them to construct the *affinity matrix* to encode the spacial coherency among feature points. Then the optimal one-to-one correspondence can be found to maximize spatial coherency. With those correspondences, a rigid pairwise alignment between frames can be found.

Global Optimization for Alignment Pruning. Pairwise alignments computed through feature matching could be affected by both availability of features and correctness of their matching computation. When the inter-frame overlap is small, even with well developed local matching algorithms, these alignments are still often inevitably unreliable. In many existing reconstruction systems [7,24,1,3,4,5], global optimization based on loop closure is adopted to eliminate accumulative error. However, such a strategy is based on an assumption that the initial alignments are good enough. If local alignments are often unreliable, this assumption no longer holds, and hence, the reliability of these algorithms suffers. Our idea is to keep multiple potential pairwise alignments between each pair of frames, then apply a global pruning to identify correct alignments and retrieve true loop closure, and then reduce the accumulated error.

4. Feature-guided pairwise alignments

To obtain potential pairwise alignments between different frames, we use both geometric and texture features and solve their correspondence to estimate the alignments between frames.

4.1. Feature extraction

Given input RGB-D frames $\{F_i\}$, we detect two types of feature points, from the depth images (point clouds) and color images, respectively. We use the NARF detector [19] to detect 3D keypoints in the depth images, and use SIFT [20] and Shi–Tomasi detectors [21] to detect 2D keypoints from the color images. Then we map these 2D keypoints to their 3D corresponding points in the point cloud (such a map can be easily obtained through a 2D-to-3D camera calibration preprocessing).

4.2. Initial feature correspondence

Before solving feature correspondence, we first pre-select a set of potentially similar keypoints by using k-nearest neighbors in the descriptor space. Specifically, we describe 2D keypoints (SIFT and Shi–Tomasi) using the SIFT descriptor [20] which encodes surrounding texture information, and describe 3D keypoints (NARF) using the FPFH descriptor [25] which encodes surrounding geometric properties such as curvature and normal. Note that while Shi–Tomasi detector is not scale-invariant, we still can calculate

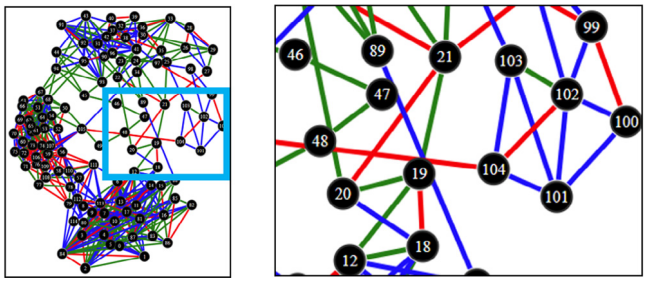


Fig. 4. Different types of features contribute useful alignments to the final reconstruction. This graph encodes the finally selected alignments (edges) between frames (nodes) in the ICL/room dataset. The red, green, and blue edges indicate alignments produced by NARF, Shi–Tomasi, and SIFT detectors, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

SIFT descriptor at Shi–Tomasi keypoint location to find correspondence between Shi–Tomasi keypoints. We expect Shi–Tomasi detector to help identify keypoints from regions with little texture and small scale changes.

For two keypoints p, q of the same type (either both are 2D or both are 3D keypoints), let $S(p), S(q)$ be the descriptor vectors of p and q. The Euclidean distance $\|S(p) - S(q)\|$ indicates their local similarity. For each keypoint, only its top-k (we empirically set $k = 1$ for texture keypoint and $k = 2$ for geometric keypoint) possible corresponding candidates are considered in the following graph matching. The output of initial correspondence is a list of potential corresponding pairs of 3D points $L = \{l_t = (p_t, q_t)\}$, $p_t \in F_i$, $q_t \in F_j$.

4.3. Feature correspondence by graph matching

Then, we construct an affinity matrix M to measure both similarity between two feature points and spatial coherency between two feature pairs. The diagonal element $M_{kk} = \|S(p) - S(q)\|$ encodes the similarity of the corresponding feature pair l_k, using the descriptor distance between $p_k \in F_i$ and $q_k \in F_j$. The off-diagonal element $M_{uv} = \delta(\|p_u - p_v\|, \|q_u - q_v\|)$ measures the Euclidean distance change between feature pairs $l_u = (p_u, q_u)$ and $l_v = (p_v, q_v)$, hence, describes the mutual consistency between l_u and l_v. Here we define $\delta(x, y)$ following [23]:

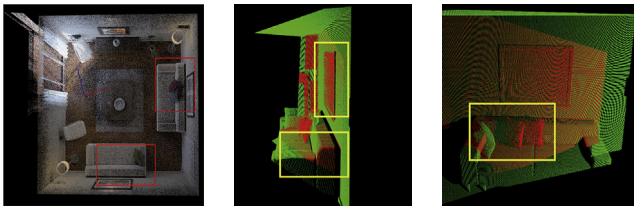
$$\delta(x, y) = \begin{cases} 4.5 - \dfrac{(x-y)^2}{2\sigma_d^2} & \text{if } \|x - y\| < 3\sigma_d \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where σ_d (we empirically set $\sigma_d = 0.01$) is a threshold to adjust how rigid the alignment should be accepted. The smaller σ_d is, the more rigid it is.

With this affinity matrix M, the node-to-node assignment that maximizes the both descriptor similarity and spatial coherency can be calculated by solving an integer quadratic problem, and we compute its approximate solution using a spectral algorithm [23].

4.4. Pairwise inter-frame alignment

With feature correspondence, we can compute the rigid transformation T_{ij} that best aligns the corresponding feature pairs from point clouds of F_i and F_j using SVD [26]. We then define an *overlap ratio* d_{ij} to evaluate how well T_{ij} aligns F_i and F_j. Two points $T_{ij} \cdot p$, $p \in F_i$ and $q \in F_j$ are considered overlapped, if $\|T_{ij} \cdot p - q\| < d_\epsilon$ ($d_\epsilon = 0.075$ in all our experiments). Let n_{ij} be the number of points in F_i that overlap with their nearest points in F_j, and $\min(|F_i|, |F_j|)$ be the smaller point size of the two point clouds. The *overlap ratio* d_{ij} then is defined as $d_{ij} = \frac{n_{ij}}{\min(|F_i|, |F_j|)}$. Only when the overlap ratio



(a) (b) (c)

Fig. 5. Incorrect pairwise alignment caused by local ambiguity. (a) Scans on the red and green box regions are similar in both geometry and color. (b) The two corresponding frames, when aligned together, have very high overlap ratio $d_{ij} > 0.85$, and are often considered locally correct. (c) But they are globally incorrect and could lead to a failure of global environment reconstruction. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$d_{ij} > \theta_P$, we consider T_{ij} to be a potential alignment between F_i and F_j. Here θ_P is a threshold ($\theta_P = 30\%$ in our experiments).

We run above algorithm on every frame pair (F_i, F_j) using all three types of features. Different features could result in different alignments. This could desirably help improve the matching reliability, as certain features are more effective in describing certain scenarios. Figs. 3 and 4 illustrate an example how multiple features contribute on the reconstruction of the ICL/room dataset. Fig. 4 visualizes on an abstract graph that after global pruning (Section 5). This graph is rendered using the javascript graph rendering tool *D3.js* [27] from our reconstruction result. Vertices represent frames and colored edges represent alignments contributed by different features (red: NARF, green: Shi–Tomasi, blue: SIFT). This figure shows that the selected alignments come from correspondences of different types of features.

The idea of using a set of complementary features can be general. If better features are learned or when reconstructing other types of multi-modality data/scans, instead of the combination of NARF, SIFT, and Shi–Tomasi detectors, other feature combinations can also be used.

Besides the fact that different features behave differently in different scenarios, pairwise alignments that produce best local match may not always be correct. Greedy strategies that pick the best local alignments could fail, especially when inter-frame overlap is small and local ambiguity is significant. Fig. 5 shows such an example. The two regions in red and green boxes have similar texture and geometry. From a local aspect, it is difficult to find out that having them matched together is incorrect. Such an ambiguity can only be detected when the matching is considered globally. Therefore, we need a global pruning to select the correct matching and refine the final reconstruction.

5. Global optimization for alignment pruning

We consider all the kept pairwise inter-frame alignments in a *pose graph* $G = (V = \{v_i\}, E = \{e_{ij}^k\})$ where index k indicates the kth alignment between fragments i and j. Each node $v_i \in V$ corresponds to the frame F_i and is associated with a *pose transformation* X_i. Each X_i is a 4×4 homogeneous matrix, describing the rigid transformation of frame F_i from its original pose to the final stitched pose. Each edge e_{ij}^k corresponds to a potential pairwise alignment T_{ij}^k, which is also a 4×4 transformation matrix. Note that this pose graph is a *multi-graph*, meaning between two nodes there could be multiple edges, which correspond to alignments obtained from different feature correspondences.

5.1. Simple pose graph and its optimization

In existing reconstruction algorithms, such as *Kintinuous* [1], and *RGB-D SLAM* [28], the concept of pose graph is also used. Pose graphs in these methods, however, are simple graphs, namely, between each pair of nodes there is at most one edge. They are constructed by greedily selecting locally optimal pairwise alignment between each pair of correlated frames.

On a simple pose graph, one needs to localize each node $v_i \in V$ in the global coordinates so that its pose X_i is consistent with all the pairwise alignments defined on its incident edges. In other words, suppose edge e_{ij} corresponds to alignment T_{ij}, then ideally the final poses X_i and X_j should satisfy $T_{ij}^k = X_i^{-1} X_j$. Concatenating the pose transformations, $\mathcal{X} = \{X_i\}$, the pose graph optimization can be formulated in Eq. (2).

$$E(\mathcal{X}) = \min \sum_{i,j} f(X_i, X_j, T_{ij}) \tag{2}$$

where $f(X_i, X_j, T_{ij})$ denotes the deviation between T_{ij} and $X_i^{-1} X_j$. Specifically, f can be formulated [29] using a nonlinear least-square error function in (3).

$$f(X_i, X_j, T_{ij}) = e(X_i, X_j, T_{ij})^T \Omega_{ij} e(X_i, X_j, T_{ij}) \tag{3}$$

where $e(X_i, X_j, T_{ij}) = \phi\left[(T_{ij})^{-1} X_i^{-1} X_j\right]$ and the operator ϕ converts a 4×4 transformation matrix to a 6-dimensional vector representing the translation and rotation; Ω_{ij} is a 6×6 information matrix to weight how important each element of $e(X_i, X_j, T_{ij})$ is.

Directly solving optimization on such a pose graph, as adopted in many systems such as *Kintinuous* [1], and *RGB-D SLAM* [28], will obviously not produce correct result if local alignments are unreliable. Because those incorrect pairwise alignments will become incorrect constraints in the calculation of node poses. One example that shows directly adopting such a pose graph optimization fails is illustrated in Fig. 6 (b, c).

5.2. Our multi pose graph and its optimization

As previously discussed, we have kept multiple potential pairwise alignments between correlated frames. While from a local aspect, it is difficult to determine which one of them is correct, we now use our constructed multi pose graph G to globally detect and prune the incorrect alignments.

This alignment pruning can be formulated as extracting a sub *single graph* $G' = \{V, E'\}, E' \subset E$ from G, such that between each pair of nodes, at most one edge e'_{ij} is kept. In other words, among all the edges e_{ij}^k between vertices v_i and v_j, at most one pairwise alignment is valid. And if all of them are incorrect, they should all be discarded.

Based on this idea, the alignment pruning and pose solving can be formulated by the following optimization:

$$\begin{aligned} &\min E(\mathcal{X}, \mathcal{L}) = E_1(\mathcal{X}, \mathcal{L}) + E_2(\mathcal{L}) \\ &\quad s.t. \quad \forall l_{ij}^k \in \{0, 1\}, \end{aligned} \tag{4}$$

where $\mathcal{X} = \{X_i\}$ represent pose transformations defined on nodes, and $\mathcal{L} = \{l_{ij}^k\}$ are indicator variables defined on all the edges. $l_{ij}^k = 1$ indicates that T_{ij}^k is a valid alignment and e_{ij}^k is selected, while $l_{ij}^k = 0$ means T_{ij}^k is invalid and ignored. And the two terms are as follows:

$$E_1(\mathcal{X}, \mathcal{L}) = \sum_{ij} \sum_k l_{ij}^k f(X_i, X_j, T_{ij}^k),$$

$$E_2(\mathcal{L}) = \sum_{ij} w_{ij} (1 - \sum_k l_{ij}^k)^2,$$

where $f(X_i, X_j, T_{ij}^k)$ is defined in (3); w_{ij} are penalty weight constants (here we adaptively set $w_{ij} = 50 d_{ij}$, d_{ij} being the *overlap ratio* defined in Section 4.4).

The first term E_1 measures the consistency between pose transformations and selected pairwise transformations whose coefficients l_{ij}^k are 1. The second term E_2 penalizes the situation that we avoid selecting any edge or select more than one edge between adjacent nodes v_i and v_j. Therefore, (4) models a trade-off between (a) selecting as many pairwise transformations as possible and making pose transformations to satisfy them, and (b) ignoring incorrect pairwise transformations which violates the majority of pose transformations computed on other nodes. A globally minimized E will produce both correctly selected pairwise transformations and the accordingly refined pose transformations.

To numerically solve Problem (4), we relax the integer constraints on $\mathcal{L}$ to linear constraints. Then we can calculate the derivatives of $\mathcal{X}$ and $\mathcal{L}$, respectively, and utilize the Levenberg–Marquardt algorithm to solve $\mathcal{X}$ and $\mathcal{L}$.

5.3. Initialization using loop closure

The numerical optimization of Problem (4) converges to a local minimum. Hence, obtaining a good initialization of poses $\hat{\mathcal{X}}_0$ and indicator values $\hat{\mathcal{L}}_0$ is important.

5.3.1. Greedy initialization

As discussed previously, when the inter-frame overlap is small, pairwise frame matching is usually unreliable even with carefully designed features. Directly picking the locally optimal pairwise alignments might not be a good idea in this scenario. For example, the state-of-the-art feature-based approaches, *ORB-SLAM* [4] and *ORB-SLAM2* [5], are still error-prone. Similarly, ICP-based methods, such as [1] and [3], also often get stuck in bad local optima. Fig. 6(b) shows an example what a greedy initialization will produce from the input (a) of the ICL/room dataset.

5.3.2. Initialization through loop closure

We compute an initial guess through *loop closure*. The *assumption* is that *correct* pairwise alignments tend to be consistent with others and they can form a loop closure, namely, the composition of consecutive pairwise alignments (transformations) will be an identity matrix; while *incorrect* pairwise alignments usually violate such loop closure constraint. This constraint can be formulated as

$$C(P) = \prod_{e_{ij} \in P} T_{ij} = I, \tag{5}$$

where e_{ij} is an edge in the loop P, and T_{ij} is the corresponding pairwise alignment, and I is the identity matrix.

Based on loop closure constraint, we design our initialization algorithm as follows. First, we extract a loop P in G using depth-first search traversal. Then we validate whether the loop closure constraint is satisfied on P. Specifically, we measure the *deviation error* ϵ from $C(P)$ to the identity matrix I. The rigid transformation matrix $C(P)$ will be converted to a 3-d translation vector $\mathbf{t}$ and 3-d rotation vector $\mathbf{r}$, and then $\epsilon = \|\mathbf{t}\|^2 + 4\|\mathbf{r}\|^2$. And we empirically set the threshold

$$\kappa = \begin{cases} \sqrt{len} \times 0.0025 & \text{if loop length } len < 10 \\ len \times 0.0025 & \text{otherwise} \end{cases} \tag{6}$$

which is adaptive to the loop length. This means that we tolerate larger accumulative error when the loop is longer, but are more strict on shorter loops. If $\epsilon > \kappa$, P is invalid, then we replace an edge $e_{ij}^k \in P$ whose overlap ratio is lowest, by another edge between nodes i and j and recheck the validity of the new loop.

When a valid loop P is obtained, we add all its edges into the initial simple graph $\hat{G}_0$. Then, we remove all the *conflict edges* from G. For example, if e_{ij}^k is selected, then $\forall h$, e_{ij}^h are removed from G. We then repeat the above procedure until no more valid loop can be found in G. When there are not enough loop closures to connect all of nodes, we introduce two heuristic strategies to link remaining isolated nodes to $\hat{G}_0$ (because we cannot use loop closure to prune their pairwise alignments for such isolated nodes). (1) If the dataset is a sequential scan, we choose an alignment of highest overlap ratio from this node's sequential neighbors. (2) If the dataset is an unordered set of scans, we globally choose the alignment of the highest overlap ratio among all edges incident with this node.

From our initialization $\hat{G}_0$, we can calculate frames' initial poses $\hat{\mathcal{X}}_0$ along these selected alignment $\hat{\mathcal{L}}_0$. Fig. 6(d) illustrates the initialization result computed from (a) using this aforementioned initialization algorithm.

All of parameters in our algorithm have been summarized in Table 3.

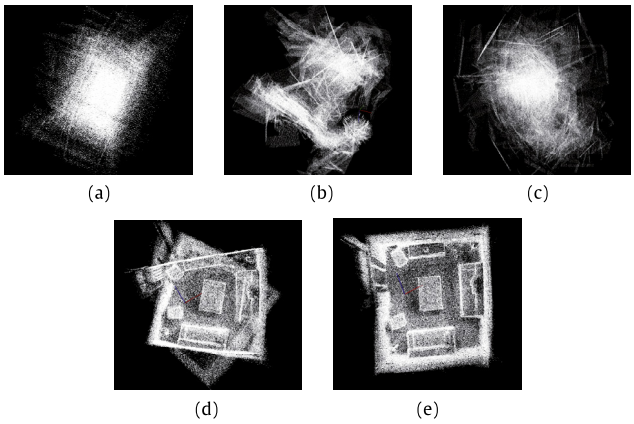


Fig. 6. Reconstruction of sparsely sampled ICL/room dataset. (a) Initial datasets with all frames rendered in global coordinates; (b) greedy stitching that selects locally optimal pairwise alignments; (c) stitching calculated by the simple pose graph optimization using (b) as an initialization; (d) our initialization result; (e) our final reconstruction after global pruning and optimization. Note that many existing algorithms adopt from greedy to simple-pose-graph-optimization strategy and hence, will lead to a result similar to (c).

6. Experimental results

We performed thorough experiments on several benchmark datasets and compared with other state-of-the-art reconstruction algorithms, both quantitatively and qualitatively.

6.1. Compared algorithms and benchmark datasets

We compare our algorithm with other state-of-the-art reconstruction systems that we are aware of. The compared algorithms include both real-time reconstruction approaches: *Kintinuous* [1], *ElasticFusion* [2], and *ORB-SLAM2* [5], and offline reconstructions approaches: *RoI* (RobustIndoor) [3], *MulF* (Multi-frame Graph matching) [15], and *COLMAP* (Structure-from-Motion) [7].

We performed experiments on two public benchmarks, the *TUM RGB-D* benchmark [30], and the *Augmented ICL-NUIM RGB-D benchmark* [3]. (1) The *TUM* benchmark contains multiple real-world scene scanning datasets with ground truth camera trajectories. We use the *fr1/room*, *fr2/desk*, and *fr3/office* datasets from the benchmark on which most other SLAM and reconstruction methods also performed evaluations and comparisons. (2) The *Augmented ICL-NUIM* benchmark provides two complete models of indoor scenes, a living room and an office room, also with ground-truth camera trajectories.

Simulating fast camera movement. Data from the above benchmarks have high scanning frame rate (30 fps) and large inter-frame overlap. The scanning movement is also slow. To compare

Table 1
Camera motions under different down-sampling factors K. $R_M/\bar{R}/R_m$ and $T_M/\bar{T}/T_m$ indicate the max, average, and min camera rotation (degrees) and translation shifts (meters), respectively.

Datasets	K	$R_M/\bar{R}/R_m$	$T_M/\bar{T}/T_m$
fr1/room	10	33.8/12.3/1.8	0.810/0.300/0.017
	8	27.4/10.1/0.6	0.634/0.244/0.011
	6	19.2/7.6/0.2	0.523/0.187/0.012
	4	14.8/5.2/0.3	0.355/0.127/0.006
	2	7.8/2.7/0.0	0.201/0.065/0.005
fr2/desk	25	48.9/6.5/1.0	1.359/0.277/0.017
	20	48.1/5.5/0.2	1.368/0.235/0.018
	15	46.3/4.3/0.0	1.340/0.180/0.008
	10	45.1/3.1/0.0	1.291/0.133/0.007
	5	45.5/1.8/0.0	1.314/0.074/0.002
	3	43.9/1.2/0.0	1.189/0.046/0.003
	2	43.2/0.9/0.0	1.219/0.034/0.001
fr3/office	25	25.7/9.0/1.8	0.922/0.286/0.027
	20	20.8/7.4/0.3	0.779/0.239/0.033
	15	18.6/5.7/0.5	0.562/0.186/0.014
	10	12.8/3.9/0.1	0.429/0.129/0.013
	5	7.3/2.1/0.0	0.234/0.069/0.006
	3	4.7/1.3/0.0	0.168/0.043/0.002
	2	3.5/0.9/0.0	0.132/0.030/0.001
ICL/room	25	49.0/6.5/1.0	1.359/0.277/0.017
	20	48.1/5.5/0.2	1.368/0.235/0.018
	15	46.3/4.3/0.0	1.340/0.180/0.008
	10	45.1/3.1/0.0	1.291/0.133/0.007
	5	45.5/1.8/0.0	1.314/0.074/0.002
	3	43.9/1.2/0.0	1.189/0.046/0.003
	2	43.2/0.9/0.0	1.219/0.034/0.001
ICL/office	25	25.7/9.0/1.8	0.922/0.286/0.027
	20	20.8/7.4/0.3	0.779/0.239/0.033
	15	18.6/5.7/0.5	0.561/0.186/0.014
	10	12.8/3.9/0.1	0.429/0.129/0.013
	5	7.3/2.1/0.0	0.234/0.069/0.006
	3	4.7/1.3/0.0	0.168/0.043/0.002
	2	3.5/0.9/0.0	0.132/0.030/0.001

performance of different algorithms under a set of scenarios with faster scanning and smaller inter-frame overlap, we reduce the sampling rate of the sequence to $1/K$, namely, we keep the first frame from every K frames. Also, to measure how rapidly the camera moves, we calculate the rotation and translation between consecutive sampled frames on the ground-truth camera trajectory. The results are documented in Table 1.

From Table 1, we have following observations: (1) The average rotations and translations shifts consistently increase with the increase of downsampling factor K. Hence, such a downsampling is a simple and reasonable approach to simulate the faster camera movement and decreased inter-frame overlap. (2) The max/min rotation and translation are sometimes not evenly increased, especially in datasets such as *fr1/room* and *ICL/room*. This is because the camera movement of the hand-hold scanning is not always stable. (3) How difficult a reconstruction is usually decided by how large the maximal rotation and translation of pose shift are. As the following experiments will show, other methods often fail when maximal pose shifts become large. The scanning data of *fr2/desk* and *ICL/room* were captured through relatively quickly moved cameras. So even when $K = 3$, the maximal shifts are big and it makes existing methods to perform badly. (4) In some datasets, such as *fr1/room*, even though the max rotations and translations are small, the actual inter-frame overlap is small. This is because the scan is performed very close to the scene/object. Such small overlap makes the sparse reconstruction highly challenging.

6.2. Evaluation schemes

Evaluating reconstruction accuracy. To perform quantitative comparison on reconstruction accuracy, we calculate the *Absolute Trajectory Error (ATE)* and *Relative Pose Error (RPE)*. *ATE* measures the camera trajectory deviation between the ground truth and the one calculated from the reconstruction. *RPE* measures pairwise relative error on poses between the ground truth and the reconstruction result. We compute *RMSE* absolute translational error for *ATE*, and the mean translational error for *RPE*, respectively. This scheme and evaluation program is publicly available [30].

Evaluating unsuccessful reconstructions. When the frame stitching is unsuccessful, different algorithms implemented different strategies to process the failure. For example, *ORB-SLAM2* [5], *RoI* [3], and *COLMAP* [7] discard frames whose correspondences and alignments could not be found correctly. *MulF* [15] also discards frames from which NARF detector cannot find enough features. *Kintinuous* [1] and *ElasticFusion* [2] will enforce the stitching of all the frames even if some alignments are incorrect. Therefore, to perform a fair comparison among different methods, we only calculate quantitative results when all the frames are stitched. If one algorithm cannot reconstruct all the frames, we use *lo- X* to indicate that it loses *X* frames, and use symbol '-' to indicate that none of the frames has been stitched.

In addition, the structure-from-motion algorithm *SfM-COLMAP* [7] uses only color images, and it reconstructs the depth information by estimating camera parameters using a multi-view model. The intrinsic parameters estimated by *COLMAP* [7], however, are different from the actual scanning sensors. Therefore, the results from *SfM-COLMAP* [7] cannot be directly compared with the ground truth for accuracy estimation. If it can stitch all of frames, we just report 'stitched'. Its evaluation is performed qualitatively (visually).

6.3. Reconstruction results and comparisons

We did the reconstruction experiments on an Intel Xeon E5-2630 with 16 GB RAM. The running time depends on how many frames need to be processed. Typically, it takes about 3 min to process 115 frames, 10 min for 287 frames, and 2 h 58 min for 1435 frames.

Table 2 documents the quantitative comparison on our and other state-of-the-art reconstruction algorithms. We can see most existing methods only work well for densely sampled data sequences, namely, datasets whose inter-frame overlaps are big. Their performances deteriorate quickly with the increase of down-sampling factor K. As discussed previously, the main reasons why most existing methods perform badly in sparse scanning are both on local matching and global composition: Locally, when inter-frame overlap is small, neither ICP-based registrations [1–3] nor single-feature-guided registrations [5,15,7] are reliable enough to provide good initialization for direct pose graph optimization or loop closure refinement. Significant amount of misalignments will easily make the direct pose graph optimization and loop closure refinement fail.

By contrast, our method produces more stable and robust reconstructions when processing data with quick camera movement and small inter-frame overlap. Adopting multiple features to generate several potential alignments and global pruning make our algorithm significantly outperform other methods in sparsely sampled scan data.

Also, our proposed global pruning strategy is general, and could work with other features. For example, when better features are learned, or when dealing with other types of multi-modality data/scans, one may choose to use other suitable complementary set of features, instead of the combination of *NARF*, *SIFT*, and *Shi–Tomasi* detectors.

Fig. 7 illustrates how reconstruction errors from our algorithm change with downsampling rates. In general, both RPE and ATE errors are positively correlated with K. Some small perturbations

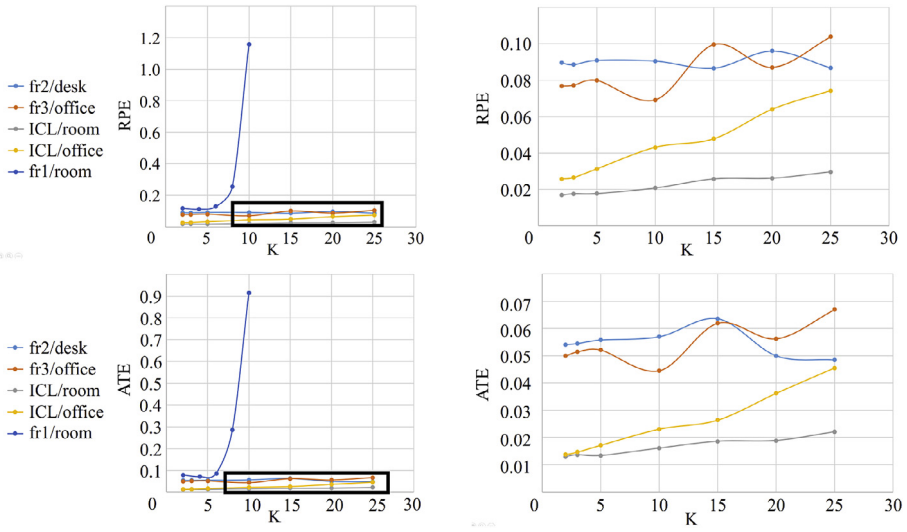


Fig. 7. Our reconstruction errors (ATE and RPE) under different downsample rates. The right images zoom in the black box region in the left images.

can be found in two *TUM* datasets, this is due to that some relatively important frames happen to be captured when K is big but skipped when K is small. Also, on the *fr1/room* data, the errors increase dramatically. This is because unlike other datasets, this data is scanned from a much closer distance from the camera. In this case, when camera shifts a little bit, the overlap between consecutive frames decreases much more significantly. The very small overlap between the downsampled frames makes the matching very challenging. This failure case has been illustrated in 8.

Some visual comparisons on results obtained from different reconstruction algorithms are shown in Figs. 9–11.

In summary, our algorithm greatly outperforms existing algorithms in processing data with small inter-frame overlap. Other methods can only process densely sampled scan datasets that have smaller maximum rotation and translation in consecutive pose shift, whereas our method is more stable for fast sensor movement. Typically, we can reconstruct from 1.2 fps ($K = 25$) in most indoor scenario, while most existing systems could not handle frame rates smaller than 15 fps ($K = 2$).

6.4. Reconstructing unordered RGB-D datasets

Our algorithm examines the matching between all frame pairs, and hence, can also naturally process unordered RGB-D image sets. On most tested datasets in the above benchmark, except for *fr1/room* ($K = 8, 10$), when we ignore the order of the frames, our reconstruction algorithm achieve the same results. In the *fr1/room* ($K = 8, 10$), loops are insufficient to provide good initializations. When the data are unordered, as we discussed in Section 5.3.2, we greedily select the highest scored pairwise alignments in initialization. This seems to be less reliable, compared with picking the optimal alignments from sequentially adjacent frames (when the scanning sequence is available). The performance difference on these two datasets is reported in Table 4. On the other hand, if plenty loops exist in the data, our algorithm produces the same result when the data are unordered in all these experiments.

Table 4
Comparison of reconstructions from sequential and unordered RGB-D data on fr1/room, where our method produces different results.

Datasets	K	RPE/ATE (m)	
		Sequential	Unordered
fr1/room	10	1.157/0.914	1.139/0.902
	8	0.255/0.286	0.846/0.779

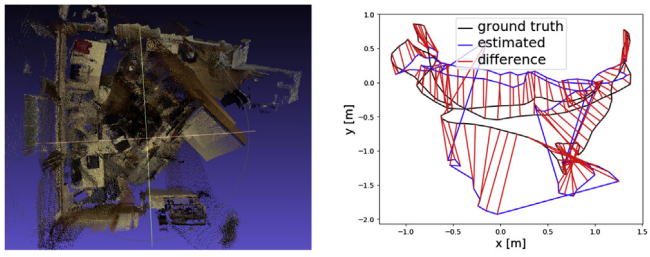
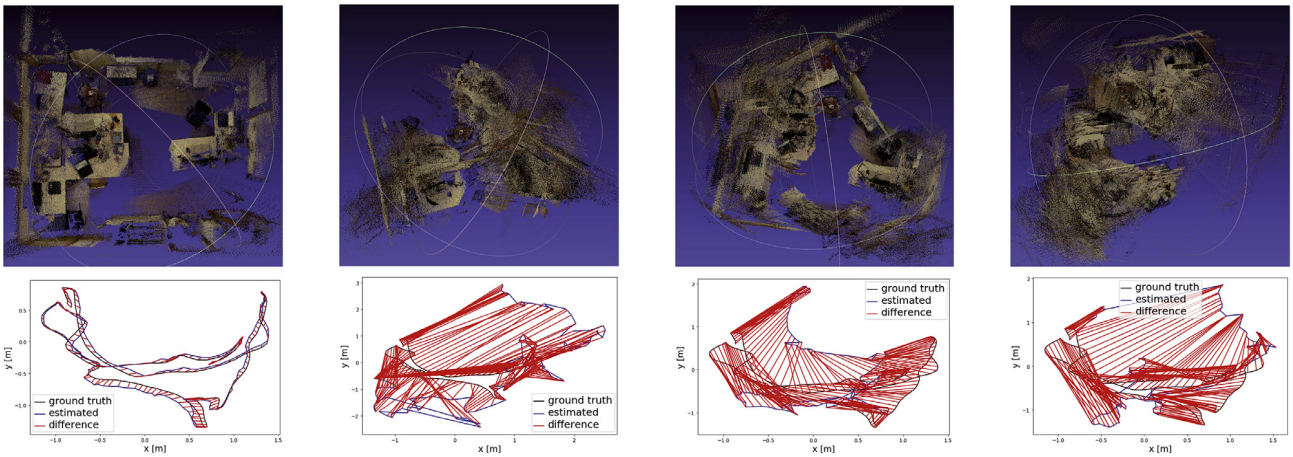


Fig. 8. A failure case: *fr1/room* with downsampling rate 1/10. The reconstruction results in two separate components, due to the lack of sufficient loops to link fragments from different components.

7. Conclusions

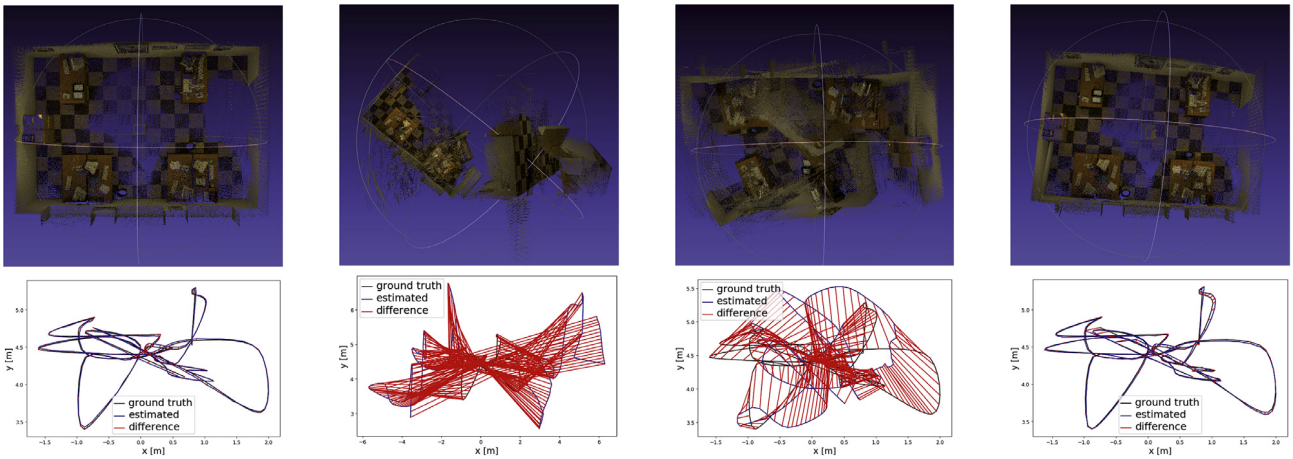
We presented a novel algorithm for 3D scene reconstruction from RGB-D data with small inter-frame overlap. We integrate multiple features to enhance the pairwise inter-frame matching under different scenarios. More importantly, we design a new global optimization algorithm to prune potential pairwise alignments and optimize frame poses. Such a global strategy significantly improves the reliability of RGB-D reconstruction when camera moves rapidly. On various public benchmark datasets we perform experiments to demonstrate the advantage of our approach over existing state-of-the-art algorithms in such scenarios.

Limitations. First, our algorithm is slow. A key efficiency bottleneck is the geometric feature extraction and pairwise matching step where we match all the $O(n^2)$ frame pairs. We could speed up the algorithm by compute these matchings in parallel. We can also improve the algorithm's efficiency and scalability by clustering frames into groups (keyframes) and hierarchically solving global pruning/optimization within each group and among different keyframes. Second, our optimization converges to local optima, and hence, still needs relatively good guess. While our initialization relies on the existence of loop closures, if there are no enough loops, the initialization degenerates to a traditional greedy strategy. We will explore more effective optimization strategy that can more stably find good global solutions.



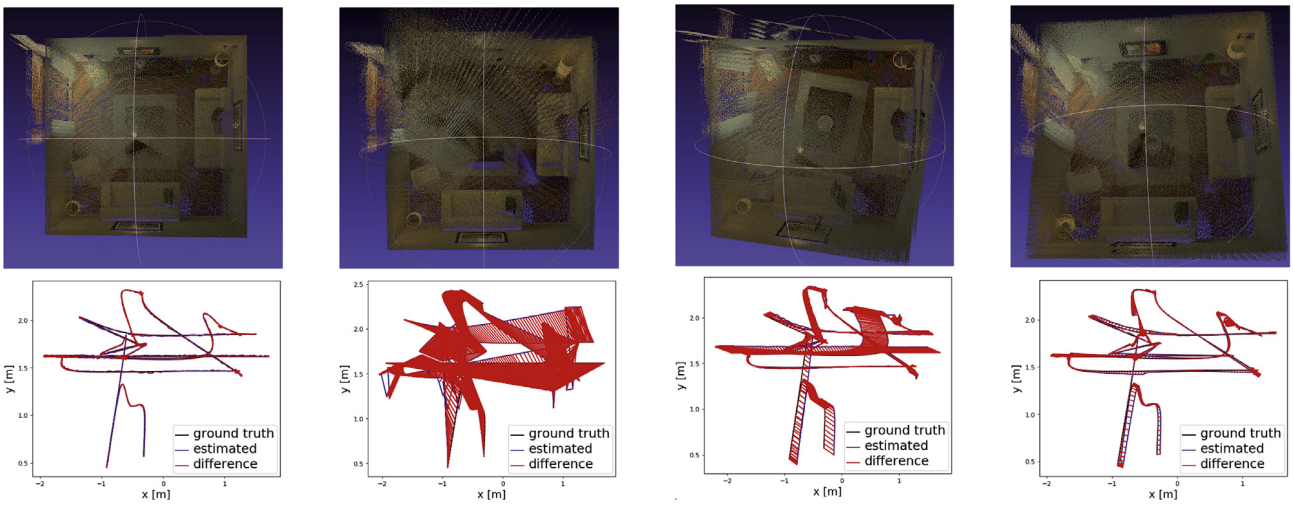
(a) Our. (b) Kintinuous. (c) ElasticFusion. (d) MulF.

Fig. 9. Reconstruction results from 1/6 downsampled fr1/room.



(a) Our. (b) Kintinuous. (c) ElasticFusion. (d) ORB-SLAM2.

Fig. 10. Reconstruction results from 1/10 downsampled ICL/office.



(a) Our. (b) Kintinuous. (c) ORB-SLAM2. (d) RobustIndoor.

Fig. 11. Reconstruction results from 1/2 downsampled ICL/room.

Table 2
Reconstruction results on TUM and ICL datasets. *1/K* is the downsample rate and *Images* indicates the number of reconstructed frames after downsampling. The number under the dataset name is the original number of frames. *lo-x* indicates *x* frames cannot be stitched and lost in reconstruction process. Symbol '–' represents none of the frames were stitched. 'all stitched' means all frames have been integrated but we cannot quantitatively compare *COLMAP* [7] with ground-truth. The best result is bolded in the table.

Datasets	K/Images	Method (Relative Pose Err/Absolute Trajectory Err) (m)						
		ORB2	Kintinuous	Elastic	RoI	MulF	COLMAP	Our
fr1/room 1352	10/136	lo-123	3.547/2.762	1.696/1.191	–	1.943/1.481	lo-61	**1.157/0.914**
	8/169	lo-146	2.858/1.896	1.452/1.121	lo-34	1.711/1.274	lo-76	**0.255/0.286**
	6/226	lo-178	2.787/1.759	1.045/0.823	–	1.959/1.334	lo-94	**0.129/0.085**
	4/338	lo-67	1.826/1.334	0.513/0.336	lo-27	2.139/1.553	lo-85	**0.112/0.071**
	2/676	**0.074/0.049**	1.154/0.953	0.381/0.291	lo-13	1.790/1.247	lo-147	0.117/0.079
fr2/desk 2174	25/87	lo-59	2.674/2.172	1.722/1.401	lo-18	1.874/1.438		**0.087/0.049**
	20/109	lo-79	2.321/1.841	1.660/1.341	lo-19	1.897/1.603		**0.096/0.050**
	15/145	lo-95	3.159/3.524	1.352/1.141	lo-14	1.730/1.355		**0.087/0.064**
	10/218	lo-144	1.953/1.449	1.169/1.117	–	1.889/1.503	stitched	**0.091/0.057**
	5/435	lo-285	1.755/1.078	0.913/0.830	lo-5	2.375/1.648		**0.091/0.056**
	3/725	lo-417	1.898/1.238	0.792/0.710	lo-4	1.075/1.221		**0.089/0.055**
	2/1087	lo-614	1.226/0.721	0.883/0.790	lo-3	2.447/1.605		**0.089/0.054**
fr3/office 2486	25/100	lo-59	2.730/1.778	1.525/1.090	lo-10	3.241/1.52	lo-27	**0.104/0.067**
	20/125	lo-38	2.416/1.718	1.217/0.830	lo-6	2.607/2.151		**0.087/0.056**
	15/166	lo-96	1.663/1.227	1.195/0.958	lo-3	2.794/1.631		**0.099/0.062**
	10/249	lo-109	2.161/1.908	0.366/0.251	lo-2	1.591/1.412	stitched	**0.069/0.044**
	5/498	**0.027/0.014**	0.280/0.285	0.193/0.153	0.092/0.051	3.331/2.202		0.079/0.052
	3/829	**0.022/0.011**	0.090/0.067	0.104/0.076	0.052/0.028	2.075/0.886		0.077/0.051
	2/1243	**0.023/0.011**	0.089/0.068	0.067/0.045	0.049/0.028	2.641/1.780		0.076/0.049
ICL/room 2870	25/115	lo-109	3.353/2.587	2.058/1.536	lo-22	lo-6	lo-38	**0.029/0.022**
	20/144	lo-127	12.567/11.939	2.000/1.465	lo-25	lo-7	lo-46	**0.026/0.019**
	15/192	lo-149	7.297/5.996	1.907/1.599	lo-192	lo-10	lo-136	**0.026/0.018**
	10/287	lo-187	2.265/1.736	1.670/1.453	lo-24	lo-14	lo-48	**0.021/0.016**
	5/574	lo-92	2.049/1.889	1.301/1.128	lo-9	lo-30	lo-35	**0.018/0.013**
	3/957	lo-138	2.728/3.039	0.493/0.397	lo-3	lo-51	lo-758	**0.017/0.013**
	2/1435	0.187/0.196	1.216/1.341	0.658/0.712	0.091/0.049	lo-76	lo-36	**0.016/0.013**
ICL/office 2538	25/101	lo-93	3.828/2.716	3.043/2.088	lo-4	lo-4	lo-52	**0.074/0.046**
	20/127	lo-117	3.329/2.294	2.835/1.767	lo-5	lo-4	lo-54	**0.064/0.036**
	15/170	lo-156	4.415/3.110	2.212/1.415	lo-7	lo-7	lo-52	**0.048/0.026**
	10/254	0.068/0.061	4.324/3.509	1.546/1.076	lo-4	lo-10	lo-27	**0.043/0.023**
	5/508	0.039/0.025	0.177/0.191	0.520/0.402	2.239/2.028	lo-20	lo-10	**0.031/0.017**
	3/846	0.038/0.026	**0.015/0.010**	0.145/0.106	2.028/1.829	lo-32	lo-12	0.026/0.014
	2/1269	0.038/0.025	**0.016/0.010**	0.093/0.066	0.051/0.019	lo-47	stitched	0.025/0.013

Table 3
Parameters table.

Parameter	Short description	Value
k	top-k nearest Euclidean distance between point pairs in descriptors space	$k = 1$ for texture feature $k = 2$ for geometric feature
σ	Graph matching threshold	0.01
d_ϵ	Distance threshold for judging inlier points between two aligned point clouds	0.075
θ_P	Overlap ratio threshold for calculating alignment candidates	30%
κ	Loop closure validation threshold which adapt to loop length *len*	$\kappa = \sqrt{len} \times 0.0025$ if $len < 10$ $\kappa = len \times 0.0025$ if $len \geq 10$

Acknowledgments

This work was partly supported by the National Science Foundation IIS-1320959. Canyu Le was supported by the National Natural Science Foundation of China 61728206, and part of his work was done while he was a visiting student at Louisiana State University.

References

[1] Whelan T, Kaess M, Johannsson H, Fallon M, Leonard JJ, McDonald J. Real-time large-scale dense rgb-d slam with volumetric fusion. Int J Robot Res 2015;34(4–5):598–626.
[2] Whelan T, Salas-Moreno RF, Glocker B, Davison AJ, Leutenegger S. Elasticfusion: Real-time dense slam and light source estimation. Int J Robot Res 2016;35(14):1697–716.
[3] Choi S, Zhou Q-Y, Koltun V. Robust reconstruction of indoor scenes. In: CVPR. 2015. p. 5556–65.
[4] Mur-Artal R, Montiel JMM, Tardos JD. Orb-slam: a versatile and accurate monocular slam system. IEEE Trans Robot 2015;31(5):1147–63.
[5] Mur-Artal R, Tardós JD. Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Trans Robot 2017.
[6] Wu C. Towards linear-time incremental structure from motion. In: Proc. 3DV. IEEE; 2013. p. 127–34.
[7] Schonberger JL, Frahm J-M. (2016) Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition, p. 4104–13.
[8] Newcombe RA, Izadi S, Hilliges O, Molyneaux D, Kim D, Davison AJ, et al. Kinectfusion: Real-time dense surface mapping and tracking. In: Mixed and augmented reality (ISMAR). 2011. p. 127–36.
[9] Mur-Artal R, Tardós JD. Fast relocalisation and loop closing in keyframe-based slam. In: Proc. ICRA. 2014. p. 846–53.
[10] Kümmerle R, Grisetti G, Strasdat H, Konolige K, Burgard W. g 2 o: A general framework for graph optimization. In: Proc. ICRA. 2011. p. 3607–13.
[11] Rublee E, Rabaud V, Konolige K, Bradski G. Orb: An efficient alternative to sift or surf. In: Intl. Conf. Computer vision (ICCV). IEEE; 2011. p. 2564–71.
[12] Triggs B, McLauchlan PF, Hartley RI, Fitzgibbon AW. Bundle adjustmenta modern synthesis. In: International workshop on vision algorithms. Springer; 1999. p. 298–372.

[13] Dai A, Nießner M, Zollhöfer M, Izadi S, Theobalt C. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. ACM Trans Graph 2017;36(3):24.
[14] Sünderhauf N, Protzel P. Switchable constraints for robust pose graph slam. In: Intelligent robots and systems (IROS), 2012 IEEE/RSJ international conference on. IEEE; 2012. p. 1879–84.
[15] Zheng S, Li B, Zhang K, Hong J, Li X. A multi-frame graph matching algorithm for low-bandwidth rgb-d slam. Comput-Aided Des 2016;78:107–17.
[16] Wang C, Guo X. Feature-based rgb-d camera pose optimization for real-time 3d reconstruction. CVM 2017;3(2):95–106.
[17] Lin S, Lai Y-K, Martin RR, Jin S, Cheng Z-Q. Color-aware surface registration. Comp Graph 2016;58:31–42.
[18] Tombari F, Salti S, Di Stefano L. Performance evaluation of 3d keypoint detectors. Int J Comput Vis 2013;102(1–3):198–220.
[19] Steder B, Rusu RB, Konolige K, Burgard W. Point feature extraction on 3d range scans taking into account object boundaries. In: Proc. ICRA. 2011. p. 2601–8.
[20] Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis 2004;60(2):91–110.
[21] Shi J, et al. Good features to track. In: Computer vision and pattern recognition (CVPR). 1994. p. 593–600.
[22] Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 1981;24(6):381–95.
[23] Leordeanu M, Hebert M. A spectral technique for correspondence problems using pairwise constraints. In: Intl. Conf. Computer vision (ICCV), Vol. 2. 2005. p. 1482–9.
[24] Endres F, Hess J, Sturm J, Cremers D, Burgard W. 3-d mapping with an rgb-d camera. IEEE Trans.Robot. 2014;30(1):177–87.
[25] Rusu RB, Blodow N, Beetz M. Fast point feature histograms (fpfh) for 3d registration. In: ICRA. 2009. p. 3212–7.
[26] Horn B. Closed-form solution of absolute orientation using unit quaternions. J Opt Soc Amer A 1987;4(4):629–42.
[27] d3js, https://d3js.org/, (Accessed: 15-02-18).
[28] Endres F, Hess J, Engelhard N, Sturm J, Cremers D, Burgard W. An evaluation of the rgb-d slam system. In: Proc. ICRA. 2012. p. 1691–6.
[29] Grisetti G, Kummerle R, Stachniss C, Burgard W. A tutorial on graph-based slam. IEEE Intell Transp Syst Mag 2010;2(4):31–43.
[30] Sturm J, Engelhard N, Endres F, Burgard W, Cremers D. A benchmark for the evaluation of rgb-d slam systems. In: Proc. IROS. 2012.