Surface Reconstruction via Fusing Sparse-Sequence of Depth Images

Long Yang, Qingan Yan, Yanping Fu, and Chunxia Xiao

Abstract—Handheld scanning using commodity depth cameras provides a flexible and low-cost manner to get 3D models. The existing methods scan a target by densely fusing all the captured depth images, yet most frames are redundant. The jittering frames inevitably embedded in handheld scanning process will cause feature blurring on the reconstructed model and even trigger the scan failure (i.e., camera tracking losing). To address these problems, in this paper, we propose a novel sparse-sequence fusion (SSF) algorithm for handheld scanning using commodity depth cameras. It first extracts related measurements for analyzing camera motion. Then based on these measurements, we progressively construct a supporting subset for the captured depth image sequence to decrease the data redundancy and the interference from jittering frames. Since SSF will reveal the intrinsic heavy noise of the original depth images, our method introduces a refinement process to eliminate the raw noise and recover geometric features for the depth images selected into the supporting subset. We finally obtain the fused result by integrating the refined depth images into the truncated signed distance field (TSDF) of the target. Multiple comparison experiments are conducted and the results verify the feasibility and validity of SSF for handheld scanning with a commodity depth camera.

Index Terms—depth image refinement, handheld scanning, sparse-sequence fusion, surface reconstruction, supporting subset.

1 Introduction

Commodity depth cameras (e.g., Microsoft Kinect [1]) open up a new way to capture 3D models. Unlike the conventional optical scanner with special scanning set-up [2], [3], commodity depth cameras make 3D scanning flexible and accessible to general users with low-cost [4], [5]. Especially, the handheld scanning manner could capture the models which are inconvenient to be scanned by a fixed scanning platform because of their weight, volume or special position. For example, the big and heavy Stanford sculpture group in [6], the relief on a large wall (Fig. 12) and a tree-trunk (Fig. 9) could be reconstructed by handheld scanning using a commodity depth camera.

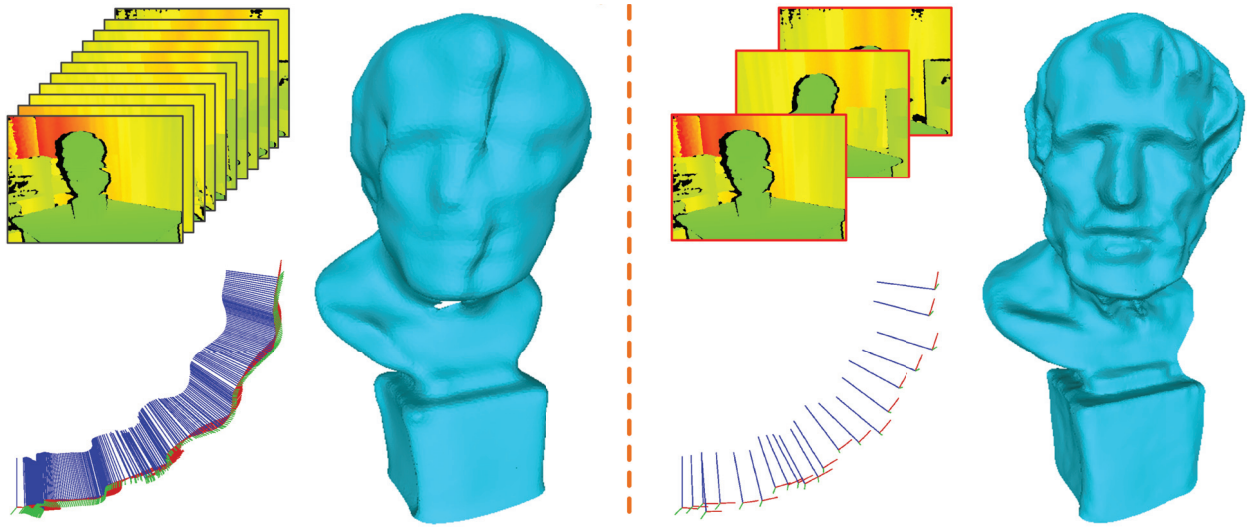
Following the fundamental pipeline of 3D reconstruction from range images, KinectFusion [7] takes frame-to-model registration to align an input depth image and incrementally integrates the aligned depth images into the fused target. Since the registration and fusion computations are loaded on GPU, KinectFusion makes real-time handheld scanning with commodity depth cameras feasible and obtains impressive reconstructed results. Users can hold a depth camera and roam around a target to get its 3D model [8], [9]. Handheld scanning by commodity depth cameras is expected to provide abundant 3D models for computer graphics community. Recently, a number of research works acquire 3D objects or scenes using commodity depth cameras based on Kinectfusion [8], [9], [10], [11], [12], [13], [14].

Although handheld scanning with commodity depth cameras has made large progress in terms of its flexibility and the real-time performance, it still contains some drawbacks. The bottlenecks mainly exist in two aspects: (1) A large number of redundant depth images are incorporated into the data fusion. To scan a moderate size object, it has to integrate nearly a thousand frames which are mostly unnecessary. (2) The jittering frames of handheld scanning might blur the geometric features of a scanned surface and even trigger the failure of camera tracking. An example is shown in Fig. 1(a), where the geometric features on the reconstructed model are smoothed and even the face is distorted.

KinectFusion scans a target with the high frame-rate for successive visual tracking. Densely fusing all the captured frames benefits denoising a single depth surface, but it involves heavy scene redundancy between consecutive viewpoints. Since the assumption of low-speed and stable camera motion [15] cannot be guaranteed for handheld scanning and the fusion will be reset once the camera tracking fails, in practice, users have to try many times to finish scanning an object. So far as we know, there is no existing work which attempts to cut down the redundant frames and generate pleasing results for handheld scanning by using commodity depth cameras.

In this paper, we present a new sparse-sequence fusion (SSF) algorithm, which is based on the extracted supporting subset from the captured depth image sequence, for handheld 3D scanning with commodity depth cameras. A unified objective function is devised to screen out the supporting depth images meanwhile filter both redundant and jittering frames. In addition, we introduce a refinement operation for the selected depth images. This refinement benefits the reconstructed result of SSF. The main contributions of this paper include:

- *L. Yang is with the Computer School, Wuhan University, Wuhan, Hubei 430072, China, and the College of Information Engineering, Northwest A&F University, Yangling, Shaanxi 712100, China. E-mail: yanglong@whu.edu.cn.*
- *Q. Yan, Y. Fu, and C. Xiao are with the State Key Lab of Software Engineering, Computer School, Wuhan University, Wuhan, Hubei 430072, China. E-mail: {Yanqingan, ypfu, cxxiao}@whu.edu.cn.*



(a) dense-sequence fusion (b) sparse-sequence fusion

Fig. 1. Depth sequence fusion of handheld scanning with a commodity depth camera. In (a), the left-top is the input depth image sequence, the left-bottom is camera trajectory of dense-sequence fusion, the right part is the reconstructed surface of dense-sequence fusion. The counterparts in (b) are sparse depth sequence, camera trajectory of SSF and the corresponding reconstruction result, respectively.

- Proposing a unified objective function to construct the supporting subset. We extract several effective measurements to describe the dynamic state of camera motion. The sparse expression problem of original depth sequence is solved by the online analysis of these measurements.
- Eliminating the dependence of dense-sequence fusion by introducing a refinement process for the selected depth images. It denoises depth images in the supporting subset meanwhile recovers their geometric features.

The remainder of this paper is organized as follows: Section 2 describes the related work of 3D reconstruction from range images. We give the overview of our algorithm in Section 3. The details of surface reconstruction from SSF and its implementation are elaborated in Sections 4 and 5, respectively. Section 6 shows the experiments and discussions. Finally, we conclude our work in Section 7.

2 Related Work

Reconstructing 3D model from multi-view range images has been widely investigated in the last two decades [16], [2], [3]. Herein we will review the related work about range scan, fusion and enhancement of the coarse depth images, and the recent progress in 3D scanning by commodity depth cameras.

The general 3D optical scanning technique contains three procedures. It first captures surface segments corresponding to the consecutive-view depth images, and then registers these segments to a unified world coordinate system, finally integrates and fuses these aligned surfaces to reconstruct the target model.

Registering multiple depth images is the basis of 3D scanning [3], [17], [18]. Iterative closest point (ICP) algorithm [19] aligns two scanned segments (i.e., estimates the variation of camera pose) via iteratively updating point-pairs and minimizing the sum of distances between all the point-pairs. KinectFusion uses Point-to-plane ICP [20] to improve the efficiency of registration. Moreover, it replaces the traditional frame-to-frame camera tracking with the frame-to-model manner, which aligns the current frame with a projected depth image on the last camera pose from the fused model. Since each referenced depth image is projected from a gradually completed unique model, frame-to-model registration could effectively reduce the drift artifact and provide a reliable estimation of camera pose for an indoor-scale scene [7], [21]. Nevertheless, the jittering frames in depth image sequence of handheld scanning will cause the failure of camera pose tracking.

Surface integration aims to remove the crack, overlap and deficiency of the aligned segments and generate a nice model. The earlier work [22] stitches multiple segments based on the Venn diagram. The method [23] clips the segments along their boundaries and merges them to be a complete mesh surface. KinectFusion employs volumetric range image processing (VRIP) [16] to fit the overlapped segments and utilizes the truncated signed distance field (TSDF) to represent the fused surface. Since the scanned target is embedded in a bounded volumetric space which is encoded with the TSDF of the fitted surface, it could update the implicit representation of target surface readily and reconstruct complex models robustly. However, it lacks screening mechanism for original depth images. Fusing the redundant frames and the jittering depth images will blur geometric features of the scanned target.

A single range surface acquired by a commodity depth camera inherently contains heavy noise [24]. Existing 3D scanning methods using commodity depth cameras eliminate noise via fusing the dense depth-image sequence [7]. There are several ways to enhance the surface segment of a single depth image. Most up-sampling methods [25], [26] could refine a coarse and low-resolution depth image

by interpolating depth-pixels under the guidance of a high-resolution RGB image. Shading based depth refinement [27], [28], [29] employs the shading decomposition of an aligned RGB image to enhance the corresponding depth image. It requires a reliable estimation for both illumination and albedo. It is infeasible to integrate these techniques into the real-time procedure of handheld scanning, since the unstable camera motion might cause blur artifacts on RGB images and precisely estimating albedos for different parts of a scene is intractable. A multi-scale method [30], which recovers geometric features without the assistance of an additional RGB image, could be adapted to refine the depth images captured by commodity depth cameras.

Relying on the mobility of commodity depth camera, together with the simultaneous localization and mapping (SLAM) technique on dense depth map [31], depth image fusion has been extended to large-scale scenes by means of translating and rotating the integrated TSDF cube [15], [9], [10]. Chen et al. [11] employ a hierarchical GPU data structure which compresses the generated TSDF volume to reconstruct large-scale scene with real-time high quality. Nießner et al [12] exploit voxel hashing rather than regular grid to efficiently access and update the implicit surface for large-scale scene. The saved time is then used for increasing ICP iterative times so that it improves the registration accuracy and generates faithful surfaces.

To scan a large scene reliably, the inevitable drift artifact of camera tracking should be well controlled. Zhou et al. [6] distribute the accumulated errors of the camera pose to the non-interesting parts so that the interesting regions will be reconstructed faithfully. The elastic fragment fusion [13] exploits non-rigid fusion between adjacent volumes to generate global consistent 3D scene. Fioraio et al. [14] reduce camera drift by updating the associated TSDFs between two adjacent sub-volumes. Recently, Xu et al. [32] explore the direction of automatic robot scanning via online scene analysis based on KinectFusion. Zhang et al. [33] integrate structural information from online analysis to enhance the reconstruction of indoor scenes. These methods have not investigated scanning reconstruction from sparse depth image sequence. The aforementioned two drawbacks of handheld scanning in Section 1 still exist. Without decimating the captured depth images, more redundant frames need to be saved and fused when a large-scale scene is scanned.

Unlike the existing approaches, our method explores surface reconstruction via SSF for handheld scanning 3D objects using a commodity depth camera.

3 Overview

3.1 Problem Statement

The basic setting of our problem is handheld scanning using a commodity depth camera. Our goal is to realize 3D reconstruction of SSF. The core problem is how to decrease the redundant depth images and simultaneously exclude the jittering frames from the original sequence. Moreover, SSF will reveal heavy noise of the raw depth images. We should generate noise-free results in spite of using less depth images.

Supporting subset. To achieve SSF, we exploit a supporting subset to represent the original depth image sequence. We do not intend to give a mathematical definition of the supporting subset. But its essential properties are provided. It should effectively decouple the supporting depth images, the jittering frames, as well as the redundant frames. The supporting subset should cover all views of the scanned target included in the original sequence. Our supporting subset will balance between the sufficiency of scanning view and the sparseness of original sequence.

Single-frame refinement. Dense fusion removes the heavy noises of the raw depth images relying on abundantly averaging the target's TSDF [7]. Sparse fusion integrates only a few depth images so that the reconstructed model will be noise-contaminated. We introduce a refinement process to improve the quality of each selected depth image in the supporting subset. It will break the dependence of dense-sequence fusion. With the refined depth images, SSF does not depend on dense fusion to eliminate noise anymore.

3.2 Algorithm Pipeline

For captured depth image sequence, we first present a target-oriented weighted ICP (WICP) to improve the accuracy of target registration. We then introduce a new module (procedure (c) in Fig. 2) to construct supporting subset for the captured depth image sequence. It provides a sparse and stable depth image subsequence for scanning fusion. After that, a real-time denoising in combination with a feature recovering operation is designed to refine the selected depth images. Finally, the refined depth images are integrated into the target's TSDF to obtain the reconstructed model. The overall pipeline of our algorithm is depicted in Fig. 2.

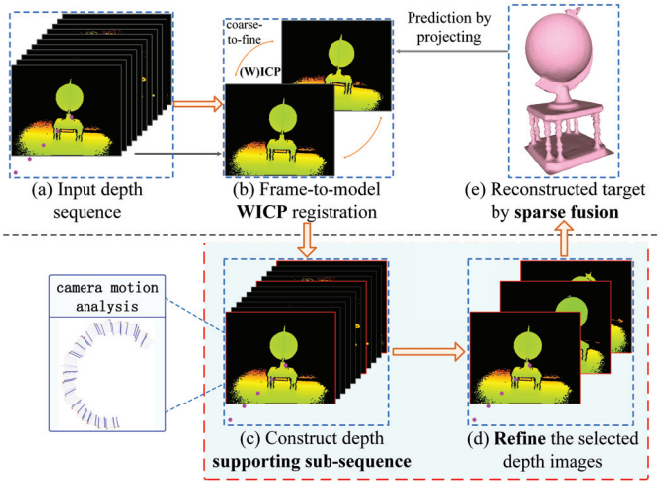


Fig. 2. The pipeline of our SSF algorithm. (a) is the input depth image sequence. A distance-weighted WICP algorithm is designed for the target-oriented frame-to-model registration in (b). We introduce module (c) to construct the supporting subset and module (d) to refine the selected depth images. (e) is the reconstructed target via SSF.

4 Surface Reconstruction Using Sparse-Sequence Fusion

4.1 Preliminary

Range scan integrates a depth image (i.e., surface segment) into the target's TSDF based on its corresponding camera pose. Given a depth image sequence $D = \{d(i)|i =$

$1, \cdots, N\}$ with N consecutive views, we attempt to construct a supporting subset $S = \{s(j)|j = 1, \cdots, M\}$ ($S \subset D$ and $M < N$), which could sufficiently represent the initial sequence D and cut down the redundant depth images as well as the jittering frames. We denote the camera trajectory as a set of sensor poses $\boldsymbol{CP}$, i.e.,

$$\boldsymbol{CP} = \{cp_1, cp_2, \cdots, cp_i, \cdots, cp_N\}. \quad (1)$$

Each pose cp consists of a view orientation v and a camera location l. For example, the camera's pose of the i-th frame cp_i can be expressed as

$$cp_i = (v_i, l_i). \quad (2)$$

The motion between two consecutive poses is represented by a rigid transformation matrix $\mathbf{T}$, defined as,

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{t}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}, \quad (3)$$

namely,

$$cp_i = \mathbf{T} \cdot cp_{i-1}. \quad (4)$$

The rotation transformation $\mathbf{R}$ and the translation $\mathbf{t}$ are calculated by a coarse-to-fine ICP procedure between the i-th depth image and the projected depth image [7].

4.2 Construction of the supporting subset

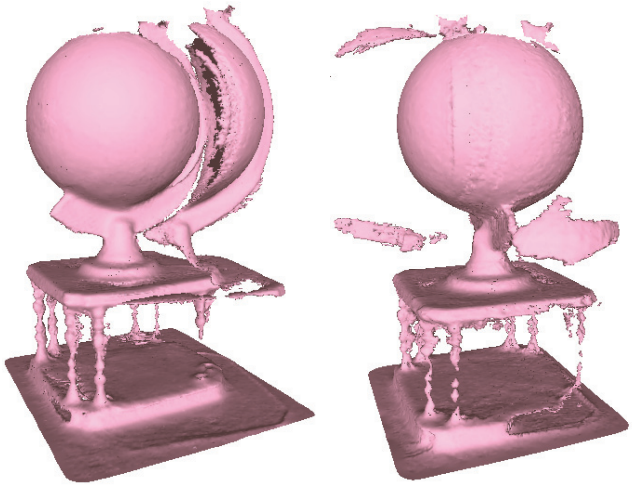
Our method reduces redundant data through sparsely sampling the original depth image sequence. Since the registration of depth image is based on the views' overlaps [7], we should hold proper redundancy to make the registration smooth rather than eliminating all the overlaps. Therefore, we construct a supporting subset to reconstruct an object with minority but sufficient depth images.

For handheld scanning, it is inapplicable to simply construct the supporting subset by straightforwardly choosing a depth image every h frames. This periodic frame fusion might trigger tracking losing or feature blurring. The examples are shown in Fig. 3. The idea that constructs the supporting subset by analyzing view coverage of the scanned target is also unadvisable for real-time handheld scanning. This strategy needs to recognize the scanned target and investigate their precise overlaps between consecutive depth images. It involves sophisticated processing of extracting 3D model from the raw depth image.

Our method resorts to more tractable quantities to approximately solve the sparse expressing problem of original sequence. Since fusing depth images is sequential and irreversible, we select the supporting depth images in order, namely frame-by-frame. We extract four measurements related to camera poses for selecting the supporting subset. Based on these measurements and their interrelationship, we construct a unified objective function to determine whether the current i-th frame should be integrated into the target's TSDF. The objective function is formulated as

$$E(i) = \lambda_1 E_{jit}(i) + \lambda_2 E_{dif}(i) + \lambda_3 E_{vel}(i) + E_{sel}(i), \quad (5)$$

where $E_{sel}(i)$, $E_{jit}(i)$, $E_{dif}(i)$ and $E_{vel}(i)$ represent the selection cost of current frame, instant variation of camera viewpoint, scene continuity and the camera motion speed, respectively. The specific definitions of these terms are as follows:



(a) h=4 (b) h=7

Fig. 3. Surface reconstruction by fixed frames fusion. (a) and (b) are the results of periodic fusion corresponding to every 4 and 7 frames, respectively. Note that the results are reconstructed by removing three times of camera tracking losing.

Selection cost of current frame. Since our goal is utilizing as few frames as possible to reconstruct an object, we introduce a switch term $E_{sel}(i)$ which controls the selection of the current i-th frame. Specifically, the term $E_{sel}(i)$ will be set to 1 if the current frame is chosen into the supporting subset, otherwise it takes 0. Thereby, the selection cost of current frame is defined as

$$E_{sel}(i) = \begin{cases} 1, & \text{if } d(i) \in S \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Utilizing this term our algorithm could practically choose the supporting depth images and decimate the redundant frames from the captured depth sequence.

Instant viewpoint change. Depth images with sudden viewpoint change (i.e. camera jittering) will cause feature blurring on the scanned model, even trigger camera tracking losing. Those jittering frames should not be fused into the final model. We consider three aspects to evaluate the instant variation of viewpoint corresponding to current depth frame. The first is the discrepancy θ_i of two camera orientations between current view v_i and its immediate predecessor v_{i-1}, defined as

$$\theta_i = \arccos(\langle v_i, v_{i-1} \rangle \,/\, (|v_i| \cdot |v_{i-1}|)), \quad (7)$$

where $\langle \boldsymbol{A}, \boldsymbol{B} \rangle$ denotes the inner product of two vectors of $\boldsymbol{A}$ and $\boldsymbol{B}$. $|\boldsymbol{A}|$ refers to the length of vector $\boldsymbol{A}$. A large θ_i means a noticeable viewpoint change with regard to camera pose of the previous frame. In practice, a camera jittering often associates with several poses' drift from local camera trajectory. Thereby, we use another discrepancy $\overline{\theta_i}$ from current view line v_i to the average view line $\overline{v_i} = avg_{j \in N_f(i)}(v_j)$ of its preceding local neighbors $N_f(i)$, i.e.,

$$\overline{\theta_i} = \arccos(\langle v_i, \overline{v_i} \rangle \,/\, (|v_i| \cdot |\overline{v_i}|)). \quad (8)$$

We assign 10 preceding frames for $N_f(i)$ in our experiments. In addition, we measure the third discrepancy $\widetilde{\theta}_i$ between current view v_i and the k-th view v_k (the k-th depth image is the latest selected frame in supporting subset S before the current frame),

$$\widetilde{\theta}_i = \arccos(\langle v_i, v_k \rangle \,/\, (|v_i| \cdot |v_k|)). \tag{9}$$

These three indicators are combined together to measure the instant viewpoint change (namely, the jittering property) of the current i-th frame,

$$E_{jit}(i) = \begin{cases} exp^{(\theta_i + \overline{\theta_i} + \widetilde{\theta_i})} - 1, & \text{if } E_{sel}(i) == 1 \\ 0, & \text{if } E_{sel}(i) == 0. \end{cases} \tag{10}$$

In Eq. (10), the jittering evaluation will be defined only if the current frame is selected into the supporting subset (i.e., $E_{sel}(i)$ takes 1). Otherwise, objective function Eq. (5) takes no account of the jittering evaluation. Fig. 3 illustrates the reconstructed results affected by jittering frames.

Scene continuity. During the down-sampling process of original sequence, sufficient scene overlap between two selected supporting frames should be maintained. Our algorithm takes the accumulated variation of camera pose as the evaluation of scene continuity. The accumulated difference $E_{dif}(i)$ from the latest selected k-th depth image to current i-th frame is defined as

$$E_{dif}(i) = \begin{cases} \sum_{j=k+1}^{i} (cp_j \ominus cp_{j-1}), & \text{if } E_{sel}(i) == 0 \\ 0, & \text{if } E_{sel}(i) == 1, \end{cases} \tag{11}$$

where the notation $\ominus$ denotes the difference of two consecutive camera poses. It regards both the camera's orientation and location, namely,

$$cp_j \ominus cp_{j-1} = s \cdot \theta_j + t_j. \tag{12}$$

Orientation change θ_j refers to Eq. (7). Location offset t_j is formulated as

$$t_j = \|l_{j-1} - l_j\|_2. \tag{13}$$

s is the tradeoff between camera's orientation and its location. We set s as 25 (one degree orientation change corresponds to 26.18mm target translation when we set the camera 1.5m away from the scanned object) in our experiments.

The term $E_{dif}(i)$ records a local accumulation of camera pose change if the current frame is abandoned (i.e., $E_{sel}(i) == 0$). Once the current i-th frame is selected into the supporting subset, this record will be reset and $E_{dif}(i)$ will be assigned to zero again. In Eq. (11), a small $E_{dif}(i)$ means higher scene continuity while a large value corresponds to lower scene continuity.

Camera motion speed. For the handheld scanning manner, camera motion speed might vary from time to time. When the camera moves with a high speed, the captured depth sequence will contain less frames. We introduce a term $E_{vel}(i)$ to evaluate the camera motion velocity

$$E_{vel}(i) = \begin{cases} \frac{\triangle t_i}{\mathfrak{m}} - \frac{\mathbb{T}}{\mathbb{M}}, & \text{if } \frac{\triangle t_i}{\mathfrak{m}} - \frac{\mathbb{T}}{\mathbb{M}} > 0, E_{sel}(i) == 0 \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where $\triangle t_i = \sum_{j=i-\mathfrak{m}}^{i} t_j$ is the accumulated distance of camera motion from the $(i-\mathfrak{m})$-th frame to the i-th frame, $\triangle t_i/\mathfrak{m}$ denotes the average distance that the camera has traversed in $\mathfrak{m}$ frames. This speed term will be calculated only if current camera speed exceeds an average speed threshold $\mathbb{T}/\mathbb{M}$ ($\mathbb{T}$ and $\mathbb{M}$ are the total distance of camera motion and the corresponding frame number respectively) and the current i-th frame is not selected into the supporting subset S.

In essence, the opposite relation implied in Eq. (5) is that both scene continuity and camera speed compete with sequence sparsity. According to the definitions of Eqs. (11) and (14), $E_{dif}(i)$ and $E_{vel}(i)$ will be accumulated if the current i-th frame is rejected, otherwise they will be reset to zero. Moreover, the term $E_{jit}(i)$ of viewpoint change accounts for the jittering evaluation of current camera pose. Therefore, the selection of current frame is intrinsically associated with local dynamical pose evaluation (i.e., the scene continuity, the camera motion speed, as well as the instant viewpoint change).

If the current i-th frame is selected our method will refine its surface segment. Then the refined surface will be fused by integrating it to the target's TSDF. Otherwise the current i-th frame will be excluded. Our algorithm successively iterates this process for all the captured depth images. It will gradually produce a supporting subset meanwhile progressively reconstruct the target via SSF.

4.3 Refinement of a selected depth image

Fusion of the selected supporting depth images greatly reduces data redundancy. However, the intrinsic noise will rise. Due to the coupled noise and geometric features on the coarse depth surface, as shown in Fig. 4(a), it is a challenge to effectively denoise a depth surface without abrading its geometric features [34], [35]. Although the method [30] could enhance geometric features, it will amplify the raw noise if it directly works on the initial depth image surface.

(a) raw depth surface (b) filtered surface (c) recovered surface

Fig. 4. Refinement of a depth frame surface. A raw depth surface segment (a) with heavy noise is filtered by a normal dissimilarity constrained filter (b) and refined by a multi-scale feature recovery (c).

We introduce a module for refining each supporting depth image. It effectively combines a feature-preserving denoising and a multi-scale feature-recovering operations.

Given a selected depth image $d(s)$, we first perform a denoising like two-step filtering [34], which works on both normal and position of each point. The normal n_i of a point p_i is updated by a bilateral normal filter $n_i^{'} = f_1(n_i)$ with the normal dissimilarity constraint [36], [37]. The position filter, $p_i^{'} = f_2(p_i)$, updates each point along its normal.

We take 8 iterations for the normal filtering and 2 times position filtering for each selected depth image surface. It could effectively denoise a surface segment while preserves its geometric features as much as possible. Fig. 4(b) shows the denoised result of a depth image surface.

The initial noise is removed on the filtered depth surface. Nevertheless, some notable geometric features are also abraded. We did not aim to recover the smoothed detail features which have the same scale level with the noise. Our purpose is to recover those abraded significant geometric features and finally to generate a quality 3D model. Therefore, we adapt a multi-scale feature enhancement technique [30] for a denoised depth surface. Unlike the detail extraction method in [30], we use the normal dissimilarity constrained bilateral filter to separate each detail layer and the base surface. Specifically, we perform 3 times of filtering and obtain three detail layers, namely,

$$p_i^{r+1} = f_2(p_i^r), r = 0, 1, 2, \tag{15}$$

$$lod_i^{r+1} = \langle (p_i^r - p_i^{r+1}), n_i^{r+1} \rangle, \tag{16}$$

where p_i^0 denotes point p_i on the initial surface, p_i^3 is the corresponding point on base surface, lod_i^{r+1} is the $(r+1)$-th level of detail for point p_i, n_i^{r+1} is the normal of point p_i^{r+1} (p_i after r times filtering). Starting from the base surface we recover the geometric features following:

$$\rho_i^r = \rho_i^{r+1} + 2.0 \cdot lod_i^{r+1} \cdot n_i^{r+1}, r = 2, 1, 0. \tag{17}$$

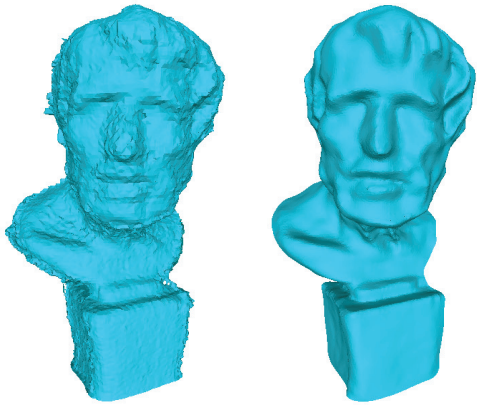
Here, ρ_i^3 is the corresponding point of p_i on the base surface (i.e., $\rho_i^3 = p_i^3$), point ρ_i^0 is the updated point of p_i on the recovered surface.

Single-frame refinement stated above is performed on the surface segment while the scanning fusion takes depth images as input. Therefore, to obtain the refined depth image, we transfer the refinement of a surface segment to the update of the corresponding depth image. Specifically, the involved geometric offset of point p_i along its normal during the refinement process will be transferred to the depth variation of corresponding pixel z_i

$$z_i' = |\rho_i^0 - p_i^0|_{\overrightarrow{z}} + z_i, \tag{18}$$

where $|\boldsymbol{A}|_{\overrightarrow{z}}$ denotes the projection of vector $\boldsymbol{A}$ along the depth direction $\overrightarrow{z}$ (i.e., camera orientation v_s). The updated depth image $d'(s)$ will participate in the sparse fusion process.

The normal dissimilarity constraint used in the filters and the detail extracting process gradually consolidates sharp features, while the multi-scale enhancement recovers the abraded notable features. A recovered depth image surface is given in Fig. 4(c). With the refined supporting depth images, our SSF could produce faithful reconstructed model. Fig. 5 shows the results of SSF with and without the refinement operation respectively.



(a) (b)

Fig. 5. Comparison of the reconstructed results. (a) is the result of SSF without depth image refinement. (b) is the sparse fused result with refined depth images. Note that both results are obtained based on WICP registration which is presented in Section 4.4.

4.4 WICP

Frame-to-model ICP algorithm is sufficient for aligning an indoor scale target [7], [21]. However, to scan a general object, there is still space to improve the registration accuracy. If a depth image contains a large proportion of background, the distant noisy depth values might dominate the ICP registration since that the accuracy degrades as the sensed depth increase [24] and that the ICP is essentially a least square process [38]. This will disturb camera pose estimation and further blur geometric features on the scanned model, see an illustration in Fig. 6.

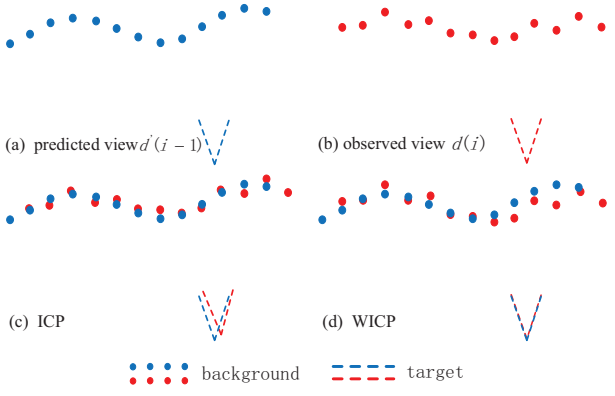


Fig. 6. 2D schematic diagram of WICP (top-view). (a) is a projected depth image $d^{'}(i-1)$ from the fused target on the last camera pose cp_{i-1}. The captured current frame $d(i)$ with heavy noise on background region is shown in (b). (c) is the ICP registration, which is governed by the distant background pixels and the target's geometric feature is blurred. The result of WICP shown in (d) overcomes this disadvantage.

We devise a weight for each pixel participated in ICP process to improve the alignment accuracy for the scanned

target. In general, the target appeared near will have small depth value in the captured depth image. We take a steep attenuated function $w(z_i)/W$ as the weight of a valid depth pixel z_i. $w(z_i)$ shown in Fig. 7 is defined as

$$w(z_i) = 1/(1 + exp^{\alpha\cdot(z_i-\beta)}), \tag{19}$$

where α is the scaling coefficient, β is the separating depth between target and background, and $W = \max\{w(z_j)|j = 1, \cdots, \mathbb{N}\}$ is the normalization coefficient, $\mathbb{N}$ is the number of the valid depth pixels. Those pixels whose depth is less than β will get a large weight while the weights of distant pixels (depth larger than β) will decrease drastically. Distance weight improves the registration accuracy of the scanned target. A comparison result is shown in Fig. 8.

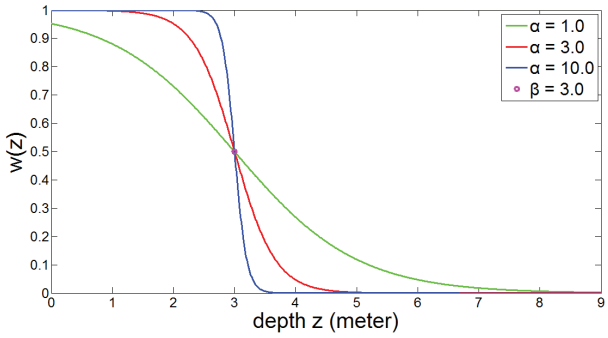
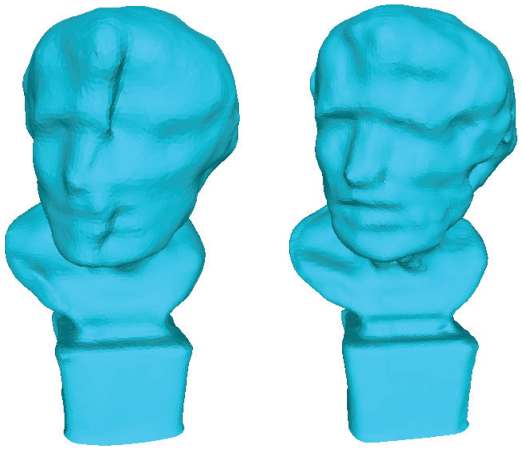


Fig. 7. Distance attenuating function $w(z_i)$ for WICP. Three attenuation curves are generated when α takes 1.0, 3.0, 10.0, respectively and β equals 3.0*m*.



(a) ICP (b) WICP

Fig. 8. Comparison of ICP and WICP. (a) and (b) are created by densely fusing depth sequence using ICP and WICP respectively. Both results are generated by removing the frames which will cause camera tracking losing.

5 Implementation

For depth sequence fusion, the rigorous optimal supporting subset does not exist. Since the sequence fusion is irreversible and the subsequent depth images are unavailable in advance, we cannot globally analyze camera poses to select the supporting depth images. Consequently, our approach adopts greedy strategy to solve the selection problem for current frame. To get a reliable and jittering-free initial frame, we start our SSF when the camera heads towards the scanned target and has a stable camera state. Those progressively selected frames will form a continuous and sparse depth subsequence with stable camera motion. Utilizing these supporting depth images our method will achieve reliable SSF.

We screen the current frame depending on Eq. (5). Since the definitions of three terms ($E_{jit}(i)$, $E_{dif}(i)$, $E_{vel}(i)$) are all related to the selection cost $E_{sel}(i)$, we pre-calculate objective function Eq. (5) under the assumptions of choosing and abandoning cases respectively, and then determine the selection of the current depth image. If the choosing cost is larger than the abandoning cost our algorithm will discard the current frame, and vice versa. The specific process is demonstrated in algorithm 1.

Algorithm 1 Select a supporting depth image.

1. **input**: camera pose cp_i of the current i-th frame
2. calculate the jittering evaluation $c_1 = exp^{(\theta_i+\overline{\theta_i}+\widetilde{\theta_i})} - 1$
3. calculate scene continuity indicator $c_2 = \sum_{j=k+1}^{i}(cp_j \ominus cp_{j-1})$
4. calculate camera speed $c_3 = \triangle t_i/\mathfrak{m} - \mathbb{T}/\mathbb{M}$
5. **if** $c_3 < 0$ **then** $c_3 = 0$ **end if**
6. assume selection cost $c_4 = 1$
7. **if** $c_4 + c_1 > c_2 + c_3$
8. $\quad E_{sel}(i) = 0$
9. **else**
10. $\quad E_{sel}(i) = 1$
11. $\quad E_{dif}(i) = 0$
12. $\quad k = i$
13. **end if**
14. **output**: $E_{sel}(i)$

Eq. (5) contains three parameters, which control the contribution of each evaluation to the total objective. We give the principle to set these parameters. A large λ_1 will increase the jittering proportion which improves the probability of abandoning the jittering frame, and vice versa. Due to the competing relationship stated in Section 4.2, large values of λ_2, λ_3 are prone to choose the current frame while small values tend to discard the current frame. We empirically set λ_1, λ_3 as fixed values 15 and 10 respectively in our experiments. In Eq. (14), we take 10 frames (i.e., $\mathfrak{m}$=10) to compute the local camera motion speed.

There are two parameters α and β in Eq. (19) which govern the WICP registration. Scaling factor α controls the attenuation amplitude of the distance weight. We set a unified α with 10 for all experimental cases. Separating depth β can be directly given by users if the depth difference between target and background is explicitly known. Otherwise a two-classes depth clustering for scanned target and background can be used to get the separating parameter β. Unless otherwise specified, we assigned 3 meters to β in our experiments.

6 Experimental Results and Discussions

With a structured light depth camera Kinect v1, we test our algorithm on different types of targets under diverse

circumstances of handheld scanning. It is implemented on a laptop with an Intel I7 CPU and an Nvidia GeForce GTX 970M graphics card.

6.1 Test models

A sculptured Seneca model, as illustrated in Fig. 1, is placed on a table surrounded by a window and two office desks. It is scanned by a limited range of viewpoints. Fig. 9 demonstrates an outdoor tree-trunk. To capture the branch of this trunk, we walk around this tree a couple of laps and spirally move down the camera. Fig. 10 is the reconstruction of the Winnie. In the first row of Fig. 11, a vase in a storage room is scanned. Due to the shape symmetry, aligning different surface segments relying on the scanned target is notoriously difficult. We resort to the background depth by setting β as 5 meters to achieve registration of the vase. The second and the third rows in Fig. 11 are a printer and a chair respectively. The chair with pluri-genus is complex. It is composed of many basic shape components so that the occlusions between different parts will interfere with the scanning process. The fourth row of Fig. 11 shows the reconstructed result of a decorative globe which is placed in a large hall. We scan it with a fast camera motion speed. A scanned small bucket is shown in the last row of Fig. 11. Fig. 12 exhibits a large reconstructed relief surface. It is scanned by moving the depth camera along the background wall. Since the relief is embedded on the wall, the separating depth β is nonexistent. We set a large value of 5 meters for β to weight each depth pixel in the process of WICP. Figs. 13 and 14 correspond to the scans of a toy tyre and a mop-slot, respectively.

All the depth sequences in our experiments are captured under handheld scanning manner. For most scanned models, we give the input depth sequence, camera trajectory (camera poses of the supporting depth images are displayed by enhanced color and line width), the result of SSF and the densely fused result, respectively.

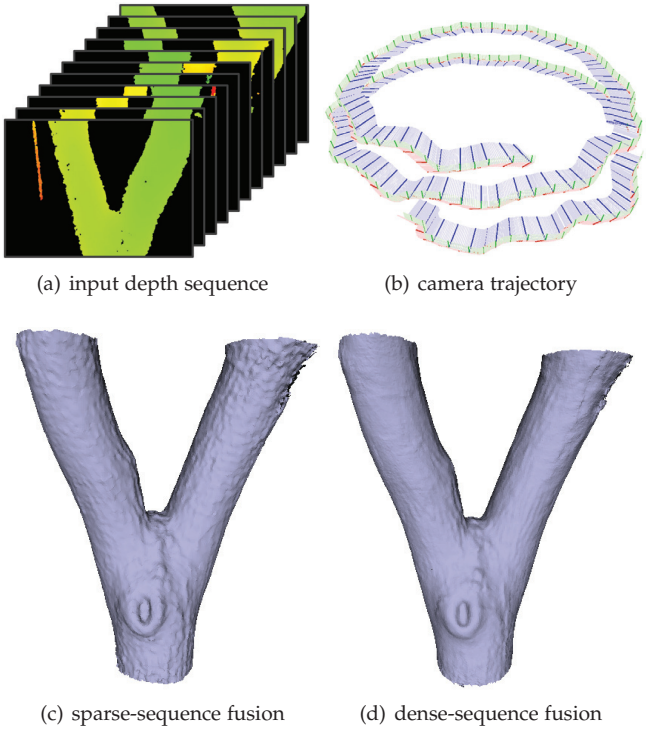
6.2 Qualitative comparison

To show the effectiveness of our method, we conduct multiple comparative experiments. The jittering frames cause camera tracking losing when Seneca, tree-trunk and globe models are scanned. An example of tracking losing for densely fusing globe sequence is shown in the accompanying video. The process of our SSF which removes the camera tracking losing is also provided.

Fig. 5 demonstrates the results of SSF by using and without using refinement operation, respectively. Refinement of the selected supporting depth images plays an important role in denoising the scanned model and recovering the geometric features for our SSF.

Comparison results of shape registration using ICP and WICP on Seneca model are given in Fig. 8. Target-oriented WICP decreases the interference of the noisy background pixels and improves the registration accuracy of the scanned target. However, the fused jittering frames still blur geometric features on the nose and hair regions in Fig. 8(b).

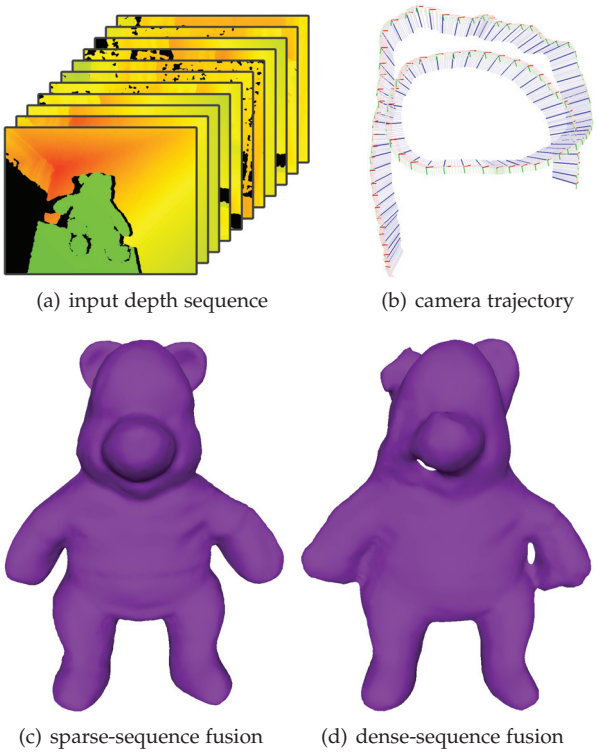
For each tested model, the results produced by both sparse and dense-sequence fusions are exhibited respectively.



(a) input depth sequence (b) camera trajectory

(c) sparse-sequence fusion (d) dense-sequence fusion

Fig. 9. Reconstruction of a real tree-trunk. (a) is the input depth image sequence. Both camera poses of the selected depth images and the discarded frames are shown together in (b). The result of SSF is shown in (c). (d) is the dense-sequence fused result by excluding the jittering frames which cause camera tracking losing.



(a) input depth sequence (b) camera trajectory

(c) sparse-sequence fusion (d) dense-sequence fusion

Fig. 10. Scan the toy Winnie. (a) is the depth image sequence. The camera trajectory is shown in (b). The result of SSF is shown in (c). (d) is the dense-sequence fused result.

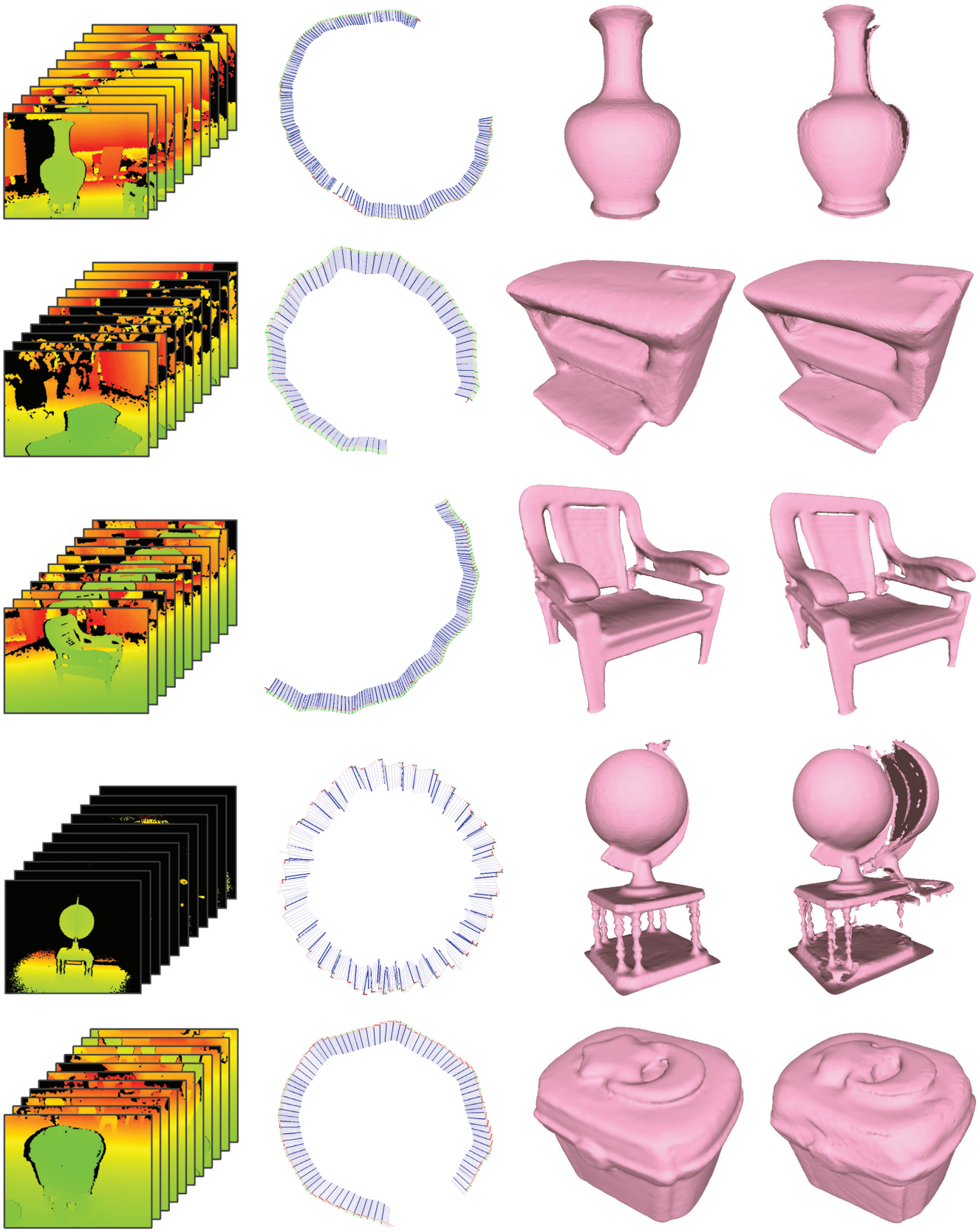


Fig. 11. Surface reconstruction via depth sequence fusion. From top to bottom, a large vase, a printer, a chair, a decorative globe and a bucket are demonstrated, respectively. For each row, the input depth image sequence, camera motion trajectory (supporting frames and the discarded frames are given together, and the supporting frames are displayed by enhanced color and line width), the result of SSF and the traditional dense-sequence fused surface are shown in order.

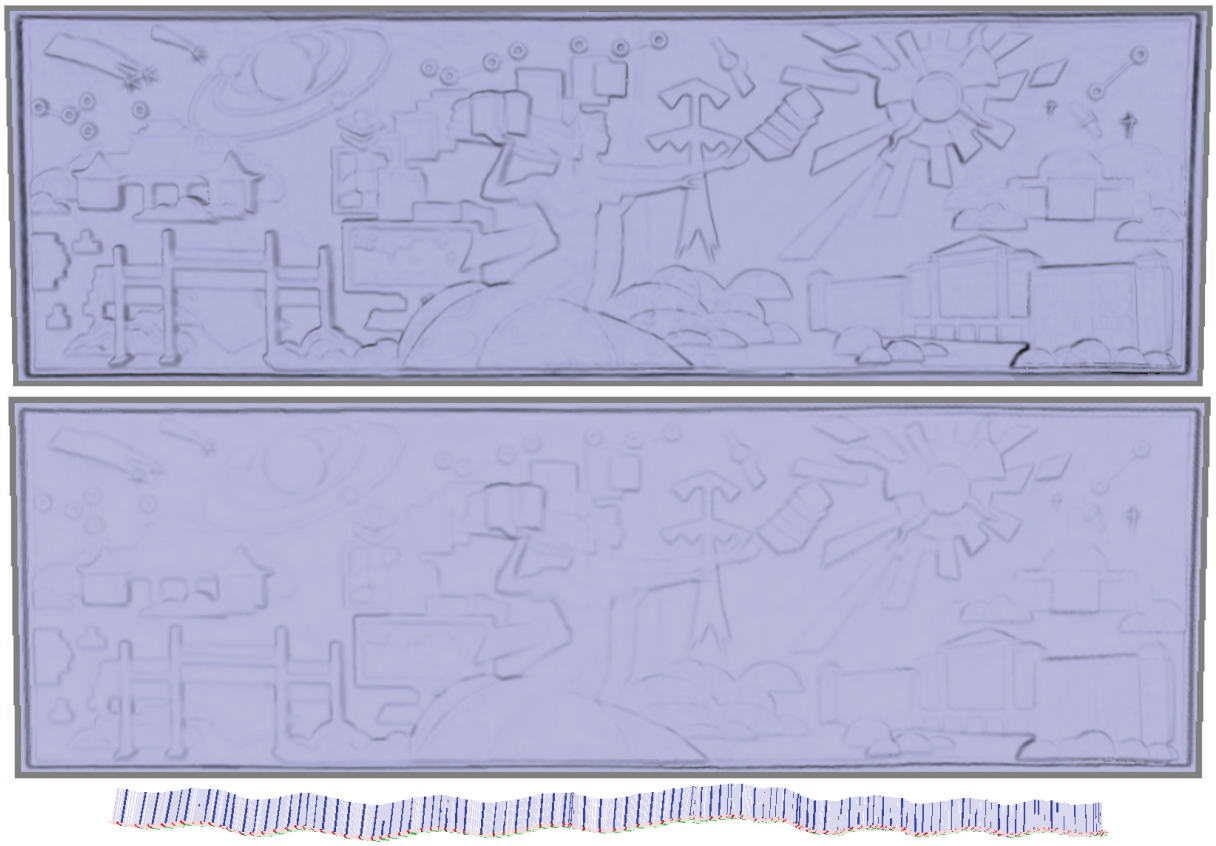


Fig. 12. Reconstruction of a relief. The top part is the result of our SSF. The middle is the dense-sequence fused result. The corresponding camera trajectory is given at the bottom.

In Fig. 1(a), due to the background noise in each depth image and the interference of jittering frames, dense-sequence fusion degrades Seneca's face and gives rise to feature drift on its nose region. Shape symmetry of the vase model aggravates the registration ambiguity. Fig. 11-1-4 (the 4-th figure of the 1-st row in Fig. 11) reveals the drawbacks of shape drift and deficiency on the densely fused result. Our method utilizes the background depth to align the scanned vase in Fig. 11-1-3 and generates a complete surface. It is always difficult to reconstruct the slim battens by aligning and integrating the captured depth images. Comparing with sparse fusion, the densely fused chair model (in Fig. 11-3-4) and mop-slot model (in Fig. 14(d)) distort their legs. Fig. 11-4-3 and Fig. 11-4-4 show the comparison results of a reconstructed globe. Our sparse fusion removes the jittering frames which will cause camera tracking losing and overcomes the drift artifact appeared in dense fusion process. Both tree-trunk model and relief surface created via sparse fusion maintain detailed geometric features, as shown in Fig. 9 and Fig. 12. The densely fused results, including the Winnie in Fig. 10(d), the bucket in Fig. 11-5-4 and the tyre in Fig. 13(d), show some blur and drift artifacts. Note that the results of dense-sequence fusion for Seneca, tree-trunk, as well as the globe models are generated by excluding the jittering frames which trigger camera tracking losing once, twice, and three times in Figs. 1(a), 9(d) and 11-4-4, respectively.

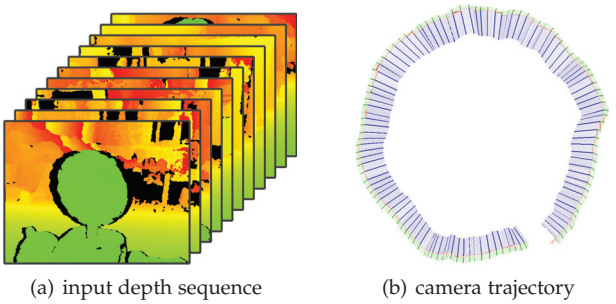
Overall, for handheld scanning by commodity depth cameras, these results verify the feasibility and validity of our SSF. Comparing with the results of dense fusion, our SSF could robustly reconstruct surfaces from these depth sequences with unstable camera motion.

6.3 Quantitative evaluation

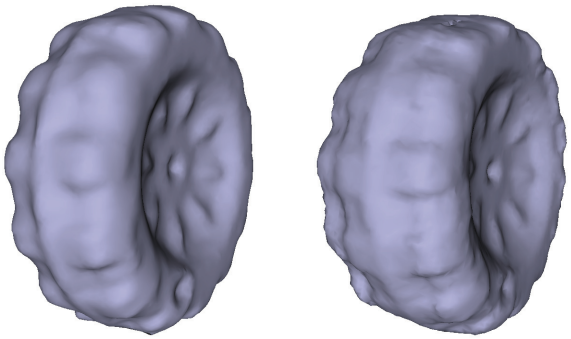
To further evaluate our method, we perform a quantitative comparison between the SSF and the dense-sequence fusion on six targets, including Seneca, printer, Winnie, bucket, tyre and mop-slot models. The ground truth surfaces of these six models are easy to obtain.

We compare the reconstructed results with the corresponding ground truth surfaces. We fix a time-of-flight (ToF) sensor and place the target object on a turntable. Then the ground truth model is produced by integrating the surface segments captured from six calibrated turning angles. The involved models are normalized into a unit cube. The Hausdorff distance between a reconstructed surface and the corresponding ground truth model is taken as the maximum error measurement. Both root mean square error (RMSE) and maximum error are reported in table 1. Errors of our results are less than those errors produced by dense-sequence fusion. The colored error plots in Fig. 15 intuitively visualize the error of each point for the reconstructed models.

Geometric features inherently demonstrates the quality of a reconstructed surface. Four different Seneca models are presented in Figs. 16(a)-(d). We cut each model with two vertical and three horizontal planes respectively to extract five

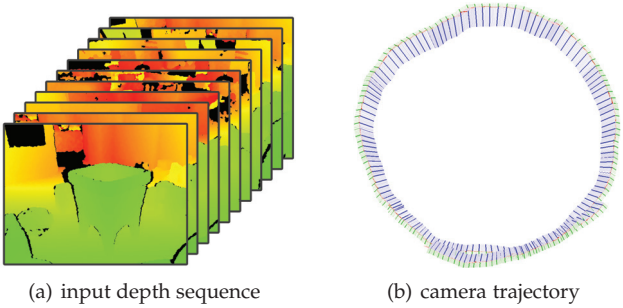


(a) input depth sequence (b) camera trajectory

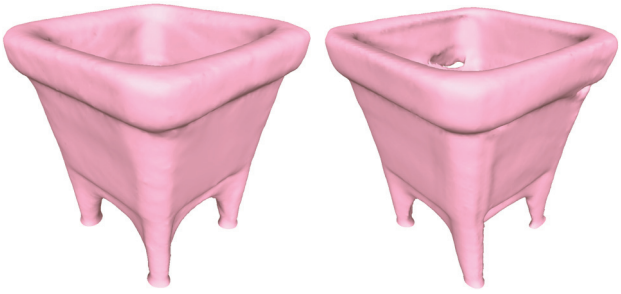


(c) sparse-sequence fusion (d) dense-sequence fusion

Fig. 13. Reconstruction of a small tyre. (a) is the depth image sequence. (b) corresponds to the camera trajectory. The result of SSF is shown in (c). (d) is the dense-sequence fused result.



(a) input depth sequence (b) camera trajectory



(c) sparse-sequence fusion (d) dense-sequence fusion

Fig. 14. Reconstruction of a mop-slot. (a) is the depth image sequence. The camera trajectory is shown in (b). The result of SSF is shown in (c). (d) is the dense-sequence fused result.

curves. Figs. 16(e)-(i) show five sets of the sectional curves. The purple curves produced by our method are closer to the ground truth curves (blue) in the feature regions. SSF generates better geometric features because of the exclusion of jittering frames and the introduced WICP as well as the refinement operation.

TABLE 1
Error evaluation of sparse and dense-sequence fusions on six models including Seneca, Winnie, printer, bucket, tyre and mop-slot.

Model	Fig.	sparse-sequence fusion		dense-sequence fusion (WICP)	
		RMSE	Max	RMSE	Max
Seneca	1	0.0175	0.1012	0.0260	0.1268
Winnie	10	0.0131	0.0545	0.0279	0.1125
printer	11-2	0.0214	0.0735	0.0230	0.0962
bucket	11-5	0.0227	0.0861	0.0254	0.0994
tyre	13	0.0067	0.0225	0.0246	0.1186
mop-slot	14	0.0136	0.0639	0.0207	0.0867

Since our method dedicates to decimating the redundant depth frames, we list the statistics including the number of depth image in original sequence and the counterpart in the selected supporting subset for all tested models in table 2. The values of parameter λ_2 and the compressing ratio of sparse-sequence are also given in table 2.

Those sequences with stable and low speed camera motion often contain high data redundancy. Our SSF realized surface reconstruction by taking 5.6%, 7.4% and 6.7% frames of the original sequences for Seneca, tree-trunk, and printer, respectively. Compressing ratios for these sequences exceed 90%. In contrast, objects with complex topology (e.g., chair model), targets with shape symmetry (vase) and sequences captured under high camera motion speed together with jittering interference (globe) have relatively low compressing ratio, see results in table 2. Even so, for all the experimental cases, our method reduces massive redundant frames and achieves at least 80% compressing ratio.

TABLE 2
Parameter λ_2 and the statistic data of experiments on the tested models, including the original frame number (N), the frame number used in sparse-fusion (M), as well as the compressing ratio of the supporting subset. The compressing ratio is defined as $(N-M)/N \cdot 100\%$.

Model	Fig.	λ_2	Original num. (N)	Sparse num. (M)	Compressing ratio(%)
Seneca	1	7.0	414	23	94.4
tree-trunk	9	8.0	1988	147	92.6
Winnie	10	13.0	900	93	89.7
vase	11-1	15.0	1041	202	80.6
printer	11-2	8.0	1101	74	93.3
chair	11-3	10.0	726	130	82.1
globe	11-4	7.0	434	70	83.9
bucket	11-5	18.0	625	84	86.6
relief	12	9.0	982	128	87.0
tyre	13	18.0	1070	113	89.4
mop-slot	14	19.0	1225	168	86.3

6.4 Efficiency analysis

SSF introduces two additional modules comparing with the traditional dense-sequence fusion. Constructing the supporting subset requires computing three measurements and one comparison operation (see Algorithm 1). This is implemented on CPU and can be processed in real-time. The main time cost we introduced is the depth image refinement. It is performed on GPU. The average time cost for refining a single depth image is 28 *ms*. It is worth to note that only the selected supporting depth images, approximately no more than 5 *fps* (frames per second) when we set the

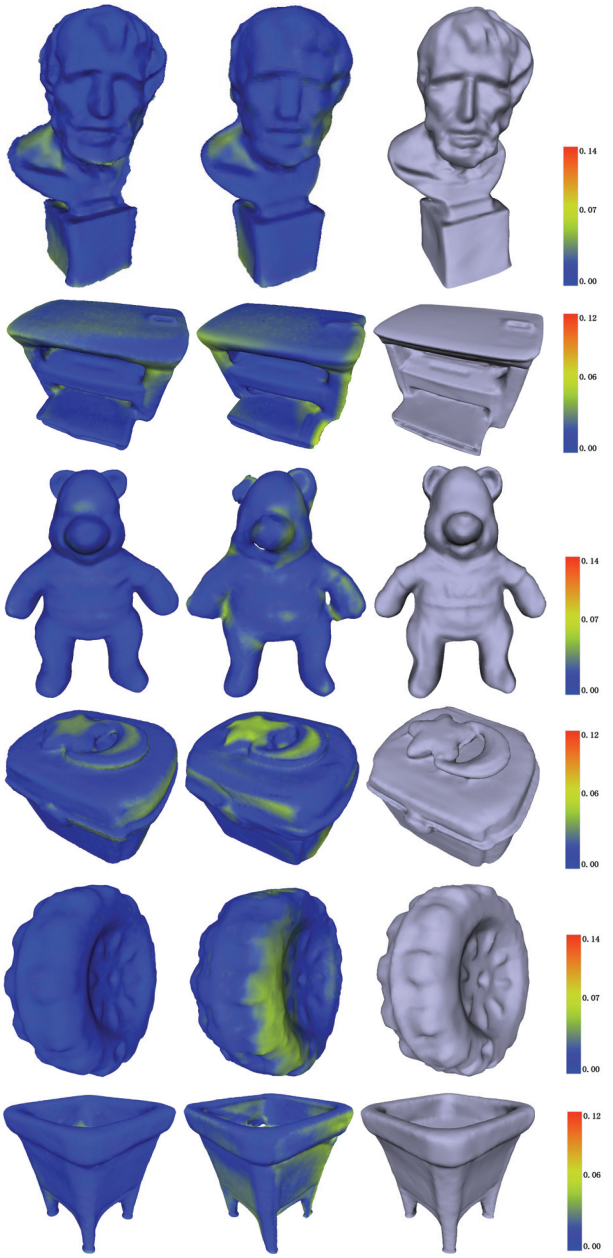


Fig. 15. Error plots for six objects. For each row, the left is the colored error plot for the result of SSF, the middle corresponds to the error plot of dense-sequence fusion and the right is the ground truth.

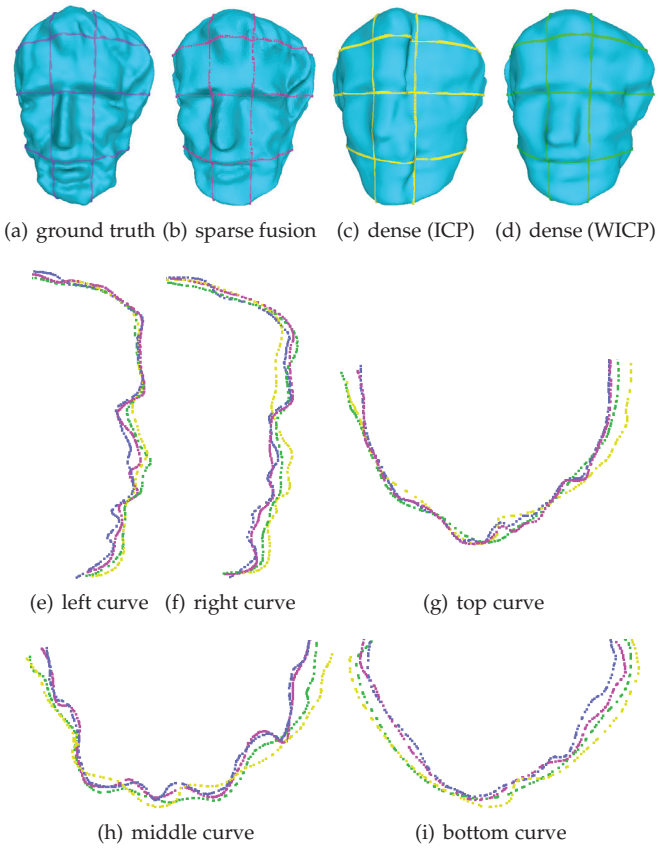


Fig. 16. Reconstruction accuracy comparison. (a), (b), (c) and (d) are ground truth, sparse fused surface, the results of dense-fusion with ICP and dense-fusion with WICP, respectively. We cut each model with two vertical and three horizontal slices and extract five curves. (e) and (f) are side-views of the left curve and right curve from the vertical slices. (g), (h) and (i) are top-views of the top, middle and bottom curves from the horizontal slices.

scanning frame rate as 24 *fps*, will be refined. In addition, our SSF will reduce many redundant depth images so that the update of TSDF on GPU for these discarded frames will be saved. Consequently, our approach is capable to run on the commodity GPU and obtain real-time performance.

6.5 Discussions

In practice of handheld scanning, camera motion following a large jittering often gets back to the adjacent pose before the jittering occurred. Hence we assume that camera poses will turn back to the scanning trajectory after a large jittering. If camera motion follows this assumption, our SSF could process the captured depth sequence and reconstruct target by employing a small number of depth images.

Registration of consecutive segments is the foundation of multi-view scanning reconstruction. Although the camera pose of a jittering frame estimated by WICP is not completely accurate, it is sufficient to identify the jittering frame from its local trajectory by the extracted jittering measurement. Therefore, the supporting subset will exclude the jittering frames. Registration between a stable frame and the progressively fused model will achieve a high accuracy so that our SSF could finally generate a quality reconstruction result.

Some objects with special shape (e.g. thin targets) are challenging to be aligned precisely. Especially when viewpoints are parallel with the thin object (see an example in Fig. 17(b)), the target pixels appeared in the captured depth image only account for a small proportion so that the accuracy of target registration will degrade drastically. Consequently, it is difficult to reconstruct these objects via automatic depth image registration.

The interesting regions of a target will often be scanned repeatedly. Therefore, abundant depth images will be captured in this case. In practice, more frames are expected to be fused for an interesting region. Since the scene continuity (in Section 4.2) is defined as the accumulated pose variations

between adjacent frames rather than the direct difference from last selected depth image to current frame, our method will select more supporting frames for a repeatedly scanned interesting region.

6.6 Limitations

There are mainly three limitations of our method: (1) The supporting subset is not necessarily an optimal sparse representation of the original depth sequence. We did not provide a rigorous theoretical analysis based on viewpoints coverage of the scanned target. Nevertheless, the supporting subset satisfies our problem setting and provides an effective balance between the sparsity of the original sequence and the continuity of the camera viewpoint. (2) For handheld scanning, the camera pose should get back to the continuous trajectory once a large jittering occurs. Our approach still suffers from the scanning reset for a drastic off-track camera motion without pose recovery. (3) Handheld scanning of the thin targets is still challenging at present. Fig. 17 shows a failure case when a bike is scanned. Although sparse fusion generates better result than dense fusion, the reconstructed surface is incomplete and has severe drift artifacts.

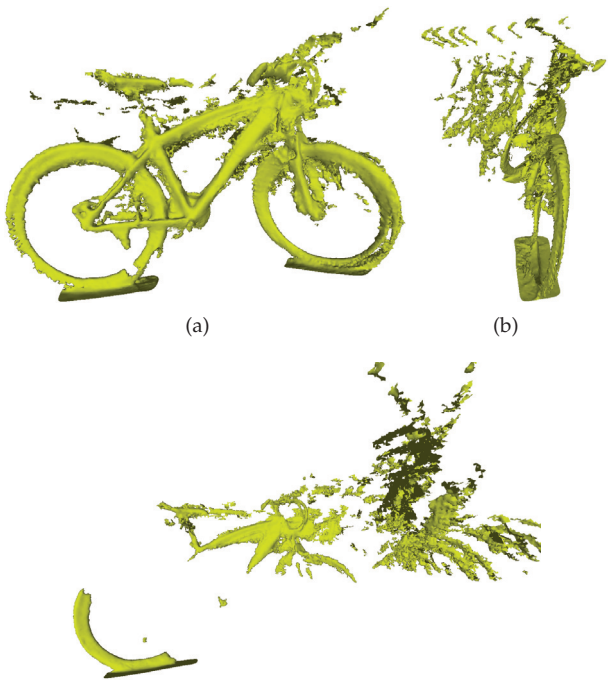


Fig. 17. Scan a bike. (a) and (b) are the side-view and front-view of our sparse-sequence fused result, respectively. (c) is the result of dense-sequence fusion.

7 Conclusion

Handheld scanning using commodity depth cameras has brought us a flexible way to get 3D models. However, directly fusing all the captured depth images cannot obtain satisfactory results. In the context of handheld scanning with commodity depth cameras, our work explores the direction of surface reconstruction by using a small number of the captured depth images for the first time. We presented a sparse-sequence fusion method in this paper. It constructs a supporting subset for the captured depth image sequence meanwhile fuses these supporting depth images sequentially. Each raw depth image selected into the supporting subset is refined by a combined operation which contains a feature-preserving surface denoising and a multi-scale geometric feature recovery. This refinement operation breaks the dependence of dense-sequence fusion. Experimental results show that our method could effectively decrease the redundant depth images and reject the interference of the jittering frames for low-cost handheld scanning.

For future work, we would like to investigate how to construct the supporting subset for original depth sequence relying on 3D content of the scanned target. Employing 3D content will enable the screening of the supporting frames more accurate. Another problem that we intend to explore is to scan and reconstruct those thin objects to enhance our SSF method.

Acknowledgments

We thank all the anonymous reviewers for their insightful comments and constructive suggestions. This work was partly supported by the NSFC (No. 61472288, 61672390), NCET (NCET-13-0441), the Fundamental Research Funds for the Central Universities (2042015kf0181, 2452015059), and the State Key Lab of Software Engineering (SKLSE-2015-A-05). Chunxia Xiao is the corresponding author.

References

[1] Microsoft, "Kinect camera," http://www.xbox.com/en-US/kinect/default.htm, 2010.
[2] D. Lanman and G. Taubin, "Build your own 3d scanner: 3d photography for beginners," in *ACM SIGGRAPH 2009 Courses*. ACM, 2009, p. 8.
[3] B. Curless, "From range scans to 3d models," *ACM SIGGRAPH Computer Graphics*, vol. 33, no. 4, pp. 38–41, 1999.
[4] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3d full human bodies using kinects," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 4, pp. 643–650, 2012.
[5] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 343–352.
[6] Q.-Y. Zhou and V. Koltun, "Dense scene reconstruction with points of interest," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 112, 2013.
[7] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136.
[8] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 2011, pp. 559–568.
[9] H. Roth and M. Vona, "Moving volume kinectfusion." in *BMVC*, 2012, pp. 1–11.
[10] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," 2012.
[11] J. Chen, D. Bautembach, and S. Izadi, "Scalable real-time volumetric surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 113:1–113:16, Jul. 2013.
[12] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 169, 2013.

[13] Q.-Y. Zhou, S. Miller, and V. Koltun, "Elastic fragments for dense scene reconstruction," in *2013 IEEE International Conference on Computer Vision*. IEEE, 2013, pp. 473–480.

[14] N. Fioraio, J. Taylor, A. Fitzgibbon, L. Di Stefano, and S. Izadi, "Large-scale and drift-free surface reconstruction using online subvolume registration," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 4475–4483.

[15] F. Heredia and R. Favier, "Kinfu large scale in pcl," http://pointclouds.org/documentation/tutorials/using_kinfu_large_scale.php, 2012.

[16] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 303–312.

[17] J. Yang, H. Li, and Y. Jia, "Go-icp: Solving 3d registration efficiently and globally optimally," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1457–1464.

[18] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, "Registration with the point cloud library: A modular framework for aligning in 3-d," *IEEE Robotics & Automation Magazine*, vol. 22, no. 4, pp. 110–124, 2015.

[19] P. BESL and N. MCKAY, "A method for registration of 3-d shapes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[20] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.

[21] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327.

[22] M. Soucy and D. Laurendeau, "Multi-resolution surface modeling from multiple range views," in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*. IEEE, 1992, pp. 348–353.

[23] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM, 1994, pp. 311–318.

[24] K. Khoshelham, "Accuracy analysis of kinect depth data," in *ISPRS workshop laser scanning*, vol. 38, no. 5, 2011, p. W12.

[25] L. Vosters, C. Varekamp, and G. de Haan, "Evaluation of efficient high quality depth upsampling methods for 3dtv," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2013, pp. 865 005–865 005.

[26] J. Yang, X. Ye, K. Li, C. Hou, and Y. Wang, "Color-guided depth recovery from rgb-d data using an adaptive autoregressive model," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3443–3458, 2014.

[27] Y. Han, J.-Y. Lee, and I. So Kweon, "High quality shape from a single rgb-d image under uncalibrated natural illumination," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1617–1624.

[28] C. Wu, M. Zollhöfer, M. Nießner, M. Stamminger, S. Izadi, and C. Theobalt, "Real-time shading-based refinement for consumer depth cameras," *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2014)*, vol. 33, p. 3, 2014.

[29] R. Or-El, G. Rosman, A. Wetzler, R. Kimmel, and A. M. Bruckstein, "Rgbd-fusion: Real-time high precision depth recovery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5407–5416.

[30] L. Yang, C. Xiao, and J. Fang, "Multi-scale geometric detail enhancement for time-varying surfaces," *Graphical Models*, vol. 76, no. 5, pp. 413–425, 2014.

[31] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 2100–2106.

[32] K. Xu, H. Huang, Y. Shi, H. Li, P. Long, J. Caichen, W. Sun, and B. Chen, "Autoscanning for coupled scene reconstruction and proactive object analysis," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 177, 2015.

[33] Y. Zhang, W. Xu, Y. Tong, and K. Zhou, "Online structure analysis for real-time indoor scene reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 5, p. 159, 2015.

[34] A. Belyaev and Y. Ohtake, "A comparison of mesh smoothing methods," in *Israel-Korea Bi-national conference on geometric modeling and computer graphics*, vol. 2. Citeseer, 2003.

[35] P.-S. Wang, X.-M. Fu, Y. Liu, X. Tong, S.-L. Liu, and B. Guo, "Rolling guidance normal filter for geometric processing," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 173, 2015.

[36] A. C. Öztireli, G. Guennebaud, and M. Gross, "Feature preserving point set surfaces based on non-linear kernel regression," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 493–501.

[37] L. Yang, Q. Yan, and C. Xiao, "Shape-controllable geometry completion for point cloud models," *The Visual Computer*, 2016, doi:10.1007/s00371-016-1208-1.

[38] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

Long Yang received the BSc degree in information and computing science from Chang'an University, China, in 2004 and the MSc degree in computer science from Northwest A&F University, China, in 2010, respectively. Currently, he is a lecturer at Northwest A&F University and working toward the PhD degree in the Computer School, Wuhan University. His research interests include computer graphics, digital geometry processing and 3D computer vision.

Qingan Yan received his BSc degree in computer science from Hubei University for Nationalities, China, and the MSc degree in computer vision and virtual reality from Southwest University of Science and Technology, China, in 2012. He is currently working toward the PhD degree in the Computer Science Department of Wuhan University. His research interests include computer vision, computer graphics and virtual reality.

Yanping Fu received the BSc degree in computer science and technology from Changchun University, in 2008 and the MSc degree in computer science and technology from Yanshan University, in 2012. Currently, he is working toward the PhD degree in the Computer School, Wuhan University. His research interests include geometry processing, 3D reconstruction and SLAM.

Chunxia Xiao received the BSc and MSc degrees from the Mathematics Department of Hunan Normal University in 1999 and 2002, respectively, and the PhD degree from the State Key Lab of CAD & CG of Zhejiang University in 2006. Currently, he is a professor at the School of Computer, Wuhan University, China. From October 2006 to April 2007, he worked as a postdoc at the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, and during February 2012 to February 2013, he visited University of California-Davis for one year. His main interests include computer graphics, computer vision and machine learning. He is a member of IEEE.