Robust 3D Reconstruction With an RGB-D Camera

Kangkan Wang, Guofeng Zhang, *Member, IEEE*, and Hujun Bao, *Member, IEEE*

***Abstract*— We present a novel 3D reconstruction approach using a low-cost RGB-D camera such as Microsoft Kinect. Compared with previous methods, our scanning system can work well in challenging cases where there are large repeated textures and significant depth missing problems. For robust registration, we propose to utilize both visual and geometry features and combine SFM technique to enhance the robustness of feature matching and camera pose estimation. In addition, a novel prior-based multicandidates RANSAC is introduced to efficiently estimate the model parameters and significantly speed up the camera pose estimation under multiple correspondence candidates. Even when serious depth missing occurs, our method still can successfully register all frames together. Loop closure also can be robustly detected and handled to eliminate the drift problem. The missing geometry can be completed by combining multiview stereo and mesh deformation techniques. A variety of challenging examples demonstrate the effectiveness of the proposed approach.**

***Index Terms*— 3D reconstruction, structure from motion, loop closure, 3D feature, global registration.**

I. Introduction

3D RECONSTRUCTION is a fundamental problem in computer vision and graphics, and can easily find applications in many areas such as movie special effects, computer game, virtual reality, and video editing. Although multiview stereo techniques [1]–[3] can be used to obtain 3D models, the robustness and accuracy is still an issue. With the development of depth sensors in recent years, there are already many research works about 3D reconstruction using low-cost depth cameras (e.g. Microsoft Kinect). For example, Cui et al. [4] described a method for 3D object scanning by aligning depth images taken around an object with a time-of-flight camera. Kinectfusion [5] was developed for accurate real-time mapping of complex and arbitrary indoor scenes in variable lighting conditions. This modeling method is popular because of its time efficiency. However, the drift problem is not well addressed in [5], which restricts its application in high-quality 3D reconstruction. Recently, Whelan et al. [6] proposed to improve Kinectfusion by addressing the loop closure problem using bag-of-words-based loop detector with SURF descriptor. However, the proposed loop closure method highly relies on the global distinctiveness of the SURF features, and requires the inlier match ratio of two frames to be above 25%, which may fail in challenging cases where there are many repeated textures or no sufficient features.

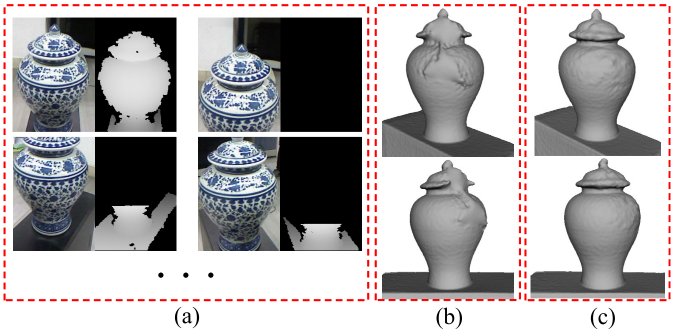


Fig. 1. Overview of our system. (a) The input RGB-D sequence of Porcelain. (b) Our reconstructed 3D model without geometry completion shown in two different views. (c) Our reconstructed 3D model with geometry completion.

In this paper, we propose a novel 3D reconstruction approach with an RGB-D camera. Compared with previous works, our system allows the RGB-D camera to freely move in challenging environments where there are many repeated textures and the captured depth maps may contain large missing areas due to range limitation or other factors (such as infrared interference for Kinect), making 3D scanning more robust and flexible. The main contributions are summarized as follows. First, we propose to combine invariant visual and geometry features and incorporate SfM technique to reliably handle fast camera motion and depth missing problem. Second, a novel global registration method with loop closure constraint is proposed which can effectively address the drift problem. Third, we propose a novel prior-based multi-candidates RANSAC algorithm which can work very well in challenging cases, where there are many repeated textures or structures. The missing geometry caused by depth missing problem also can be reliably completed by combining multi-view stereo and mesh deformation techniques. Fig. 1 shows a challenging example, where there are many repeated textures and many captured depth maps contain large missing areas. The camera motion is also rather fast (only capture 109 frames in total). Our system still faithfully recovers the complete 3D model.

Manuscript received January 24, 2014; revised July 10, 2014; accepted August 12, 2014. Date of publication September 4, 2014; date of current version October 2, 2014. This work was supported in part by the National 863 Program of China under Grant 2013AA013802, in part by the Natural Science Foundation of China under Grant 61232011 and Grant 61272048, in part by the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant 20110101130011, in part by the Fundamental Research Funds for the Central Universities under Grant 2014FZA5017, in part by the Foundation for the Author of National Excellent Doctoral Dissertation of China under Grant 201245, and in part by Huawei Technologies Company, Ltd., Shenzhen, China. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Andrew Lumsdaine.

The authors are with the State Key Laboratory of Computer-Aided Design and Computer Graphics, Zhejiang University, Hangzhou 310058, China (e-mail: wangkangkan@gmail.com; zhangguofeng@cad.zju.edu.cn; bao@cad.zju.edu.cn).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes a video. The total size of the video is 23.4 MB. Contact zhangguofeng@cad.zju.edu.cn for further questions about this work.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2014.2352851

II. Related Work

A. 3D Local Features

Many 3D local features arise in recent years due to the increasing availability of low-cost 3D depth sensors. The computation of 3D local invariant features generally contains two stages, i.e. feature detection and descriptor generation. Some approaches defined the two stages together, such as MeshDoG [7] and 3D SURF [8]. More works defined only one of them. Salti et al. [9] investigated on the effectiveness of many possible combinations between 3D detectors and descriptors. Tombari et al. [10] provided an evaluation of several 3D keypoint detectors. They evaluated the detectors using three kinds of datasets, Laser Scanner, Space Time, and Kinect datasets. Kinect datasets generally contain many holes/artifacts and stronger depth noise. Although many detectors performed rather well on the Laser Scanner and Space Time datasets, their performance obviously degraded for Kinect data. Tombari et al. [11] categorized the 3D descriptors into two classes, Signatures and Histograms. They tested the performance of some descriptors on Space Time data with satisfying result. However, the performance on noisy data remains to be investigated and improved. Until now, few works try to use the 3D features for tracking on Kinect data because of their sensitiveness to noise and outliers.

B. Multi-View Registration

Registration of multiple depth images is a significant and fundamental research topic both in computer vision and computer graphics. Strategies for registering multiple depth images can be represented by two approaches, sequential or pairwise registration and simultaneous registration. One straightforward approach of sequential registration is to register each frame to its previous frame. This method can easily accumulate registration error. To alleviate this problem, frame-to-global-frame methods [5], [12]–[15] were proposed. These methods aligned each frame independently to a global model accumulated from all previous frames. However, due to no loop closure constraint, the drift problem still occurs especially for a large scale scene.

Simultaneous registration aligns all frames at once. Nishino and Ikeuchi [16] proposed a simultaneous registration method based on an error metric computed from point-to-point distance among each frame. This method needed to search correspondence for each point in each frame with the point clouds of all fames and minimize the global error metric which is computationally expensive. Benjemaa et al. [17] and Krishnan et al. [18] proposed to register multiple 3D point sets using unit quaternions and through Optimization-on-a-Manifold respectively. Both methods required known point correspondences among overlapping parts of different point sets. However, searching point correspondences among views is exhaustive. Sharp et al. [19] and Lu et al. [20] defined a graph of pairwise registration between the neighboring views and distributed the accumulation error of pairwise registration into all frames along the graph. Pairwise registration was first estimated in geometry level. Accumulation error was then distributed on an upper level, where the error was measured in terms of the relative rotations and translations of pairwise registration. These methods did not require point correspondences between views, and can simultaneously minimize the errors of all views rapidly. However, the input data used in these methods are of high resolution and little noise, which are typically acquired by a laser range scanner or a high precision structured light 3D scanner. In addition, the manual construction of a graph is complicated and tedious under a long sequence of depth images with many closure loops. With recent advancement in the development of graph optimization, graph-based global alignment methods are popular in robotics and computer vision. Kummerle et al. [21] summarized recent works of graph-based SLAM and bundle adjustment, and proposed a general framework for graph optimization.

C. 3D Reconstruction With RGB-D Data

Many RGB-D systems have been developed due to the advent of RGB-D sensors, such as the Microsoft Kinect. Steinbrucker et al. [22] introduced a fast energy-based approach to rigidly align the RGB-D images for a static scene. Khoshelham et al. [23] presented an epipolar search method to obtain more accurate 3D correspondences and defined adaptive weights for the 3D points based on their theoretical random error to improve registration accuracy of RGB-D data. By combining both low level feature correspondences and high level plane primitives from an RGB-D camera, Dou et al. [23] improved indoor 3D reconstruction in challenging cases with insufficient image features or geometry information. Chen et al. [24] designed a compact GPU data structure for live reconstruction of large-scale scenes. Whelan et al. [25] made an extension to the Kintinuous [26] algorithm for spatially extended KinectFusion, which is able to build dense fully colored models of spatially extended extended environment in real time. The above two real-time systems will fail when the camera moves fast or the scene lacks visual/depth features. Another limitation of these real-time systems is that they are not able to solve drift problem. Methods for dealing with loop closure were proposed in [27] and [28], which detected the closure loops by frame-to-frame feature matching but relied heavily on the distinctiveness of visual features and the high inlier matching ratio. These two methods presented a similar graph-based pose optimization which was also employed in the recent work [29]. Fioraio et al. [30] detected loop closure by performing a pairwise alignment using ICP in conjunction with visual features between the first frame and the previous keyframe. However, many loop closures may be not detected using the pairwise alignment method. What's more important is that using this simple loop closure constraint is still difficult to thoroughly eliminate the drift, which is demonstrated in the experiment of Section VI. Our method can effectively eliminate the drift by detecting loop closure based on the reference feature set relating many frames. Recently, Whelan et al. [6] developed a method for handling loop closure for large scale dense RGB-D SLAM which used the bag-of-words-based loop detector [31] with SURF [32] descriptor. By taking advantage of camera pose graph optimization and non-rigid space deformation, they can obtain accurate dense

1. **Feature extraction**:
   1.1 Extract features on the current frame.
2. **Feature matching**:
   2.1 Estimate the camera pose for each frame with the proposed PMCSAC.
   2.2 Build the correspondences between features on the current frame and the reference features distributed on the previous and current subsequences.
3. **Bundle adjustment**:
   3.1 Refine the positions of reference features and camera parameters for each subsequence.
   3.2 If loop closure is detected, global refinement is applied to eliminate the drift.
4. **[Optional] Geometry completion**:
   Complete the missing geometry using the recovered depth maps by multiview stereo technique [2].
5. **Model Construction**

Fig. 2. Overview of our system.

maps with local and global consistency over large scale trajectories. However, the proposed loop closure detection method highly relies on the distinctiveness of visual features, and requires the inlier match ratio of two frames to be above 25%. In contrast, our loop closure detection method utilizes 3D information to significantly reduce matching ambiguity (only needs 3 accurate 3D-3D correspondences). In addition, depth missing and repeated texture/structure problems are not considered in their system. In contrast, with the proposed prior-based multi-candidates RANSAC algorithm, our system can handle very challenging cases where there are serious depth missing and many repeated textures/structures.

III. Overview

We propose an effective approach for robust and accurate 3D reconstruction with a handheld RGB-D camera (e.g. Kinect). Fig. 2 gives an overview of our system. There are five steps. In the first step, we detect the features on each frame using both color and geometry information. In the second step, we create a reference feature set which contains the features from the first frame initially. For each new frame, we build the correspondences between the newly extracted features and the reference features by estimating the camera pose. The matched features are merged and the unmatched features are added into the reference feature set. In the third step, we employ bundle adjustment to simultaneously optimize 3D positions of the reference features and camera poses of the subsequences for each l frames. l can be adjusted according to the frame rate. For a normal frame rate sequence, we generally set $l = 30$. If loop closure is detected, global refinement is applied to eliminate the accumulation error. In the fourth step, after all frames are registered into the reference coordinate frame, if necessary, we can use multiview stereo technique [2] to recover the depth maps, and use them to complete the missing geometry caused by depth sensor. Finally, we can construct the 3D model using either Poisson surface reconstruction method [33] or the surface reconstruction method by KinectFusion [5].

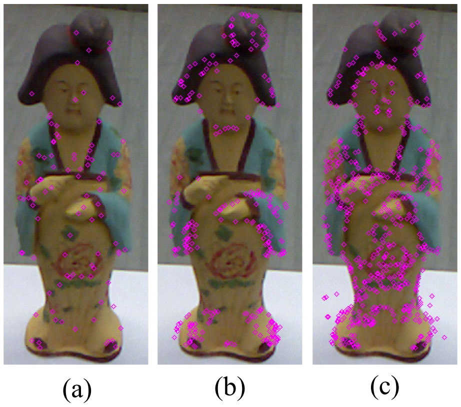


Fig. 3. Keypoint detection comparison. (a) 117 keypoints are found using SIFT. (b) 261 keypoints are found using geometry information. (c) 513 keypoints are found using both geometry and texture information. The detected keypoints are circled in red.

IV. Feature Extraction and Pose Estimation

A. Feature Extraction

Feature extraction includes two steps, keypoint detection and descriptor generation. Since our processed scene may be very complex and there are large textureless regions with strong occlusions, only using traditional feature extraction methods (e.g. SIFT) may not extract sufficient and stable features. Therefore, we propose to combine color and depth information to extract more stable features.

1) Keypoint Detection: The Harris corner detector [34] is a popular point detector due to its robustness to rotation, scale, illumination variation and image noise. By combining the surface normal and the intensity gradient, the Harris matrix can be defined as follows:

$$\mathbf{C6D}_i = \sum_{k \in W} \mathbf{G}_k \mathbf{G}_k^\top, \tag{1}$$

where $\mathbf{G}_k = (\mathbf{N}_k^\top, \mathbf{D}_k^\top)^\top = (N_x, N_y, N_z, D_x, D_y, D_z)^\top$, and W is the neighbor set of point i. $\mathbf{N}_k^\top = (N_x, N_y, N_z)$ is the surface normal of point k, and $\mathbf{D}_k^\top = (D_x, D_y, D_z)$ is the intensity gradient at point k. $\mathbf{D}_k^\top$ is orthogonal to the surface normal and points in the steepest descent direction of local intensity. The vector's magnitude indicates the rate of intensity change.

The corner strength in a point is computed using the third eigenvalue of the Harris matrix. If the corner strength of a point is larger than a threshold and that of its neighbors, we will detect the point as a corner. The number of keypoints will increase if both surface normal and texture information are used. A keypoint detection example is shown in Fig. 3. The implementation of Harris corner detection on point cloud and RGB image can be found in Point Cloud Library.[1]

[1] PCL: http://www.pointclouds.org/

2) Descriptor Generation: For each detected keypoint, we will generate a descriptor to describe it. We employ SIFT descriptor [35] as the color component of our descriptor. The geometry component is generated using FPFH [36] which describes the local geometry around a point. We first normalize the color and geometry components respectively, and then combine them to define the final feature descriptor similar to [37]:

$$f = (\alpha f_{2D}, \beta f_{3D}), \tag{2}$$

where f_{2D} is the normalized vector of color component, and f_{3D} is the normalized vector of geometry component. The color and geometry parts of the combined descriptor can be scaled by a constant (α or β) to adjust their influences on the whole descriptor. Actually, we find that good results can be obtained only using the color or geometry component in most of our examples except the extremely textureless cases. Therefore, for speedup, in many examples, we only use the SIFT descriptor as our feature descriptor.

B. Pose Estimation

By matching the 3D features of current frame with the reference features, we can obtain some 3D-3D correspondences, many of which are incorrect. Here, we propose a Prior-based Multi-candidates RANSAC, which can efficiently and reliably find the inliers and estimate the camera pose of current frame by minimizing the following energy function:

$$E(R, t) = \sum_i \|(Rg_i + t) - p_i\|^2, \tag{3}$$

where (p_i, g_i) is a pair of 3D-3D correspondence (p_i is a 3D feature on the current frame, and g_i is a reference feature). R is the rotation matrix of the current frame, and t is the translation vector. R and t transform the point from the reference coordinate frame to local coordinate frame of the camera. We call this pose estimation method as 3D-3D method. The proposed Prior-based Multi-candidates RANSAC will be described in the next section. If at least three correspondences with the reference feature set are built during feature matching, the 3D features of the current frame can be transformed into the reference coordinate frame successfully. However, Kinect only can capture valid depth information within a certain range (approximately 0.5–6m) and is easily interfered by infrared. When most depth information is not captured for current frame, it is challenging for the above registration to work well. Especially, if less than 85% of the mapped features find the corresponding reference features, we will employ another two methods (i.e. 2D-3D and 2D-2D camera pose estimation), to make the registration more robust.

First, we match the extracted 2D features on the current frame with the reference features by comparing their descriptors (only color component). With the obtained 2D-3D correspondences, the camera pose of the current frame can be estimated by minimizing the following energy function:

$$E(R, t) = \sum_i \|K(Rg_i + t) - \mathbf{x}_i\|^2, \tag{4}$$

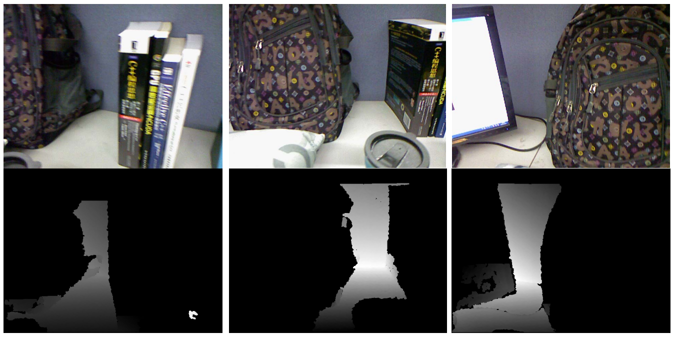


Fig. 4. Examples with significant depth missing. Top row: three selected color images. Bottom row: the corresponding depth images. The black regions are the depth missing areas.

where $(\mathbf{x}_i, g_i)$ is a pair of 2D-3D correspondence ($\mathbf{x}_i$ is a 2D feature on the current frame and g_i is a reference feature). K is the intrinsic matrix. We call this camera pose estimation method as 2D-3D method. After registering 2D features into the reference coordinate frame, we compute the ratio of the number of 3D features on the current frame that find corresponding reference features to the total number of 3D features.

Second, we match the extracted 2D features on the current frame with the 2D features on the previous frame by comparing their descriptors (only color component). Then, the relative camera pose between these two frames can be estimated using the Five-Point method [38]. The translation scale can be easily estimated by the inlier correspondences which have depth information on both frames. The rotation matrix remains the same. Finally, the camera pose for the current frame is estimated according to the relative camera pose and the camera pose of previous frame. We call this camera pose estimation method as 2D-2D method. The number ratio of 3D features that find correspondences is also computed. From the estimated poses using the above three methods, we choose the pose estimated by the method with the highest ratio.

By projecting the reference feature points to the current frame using the estimated camera pose, we can build more 2D-3D correspondences for the current frame with the reference features. For the 3D features on the current frame that have depth information, we will find their correspondences with the reference features, and the unmatched ones are added into the reference feature set. For the 2D features without depth information, their 3D positions can be triangulated with the matched 2D positions, and then added into reference feature set.

We capture a sequence of 1, 228 frames to test the effectiveness of our algorithm on depth missing data. During capture, we put the Kinect very close to the objects so that most areas of the captured frames may miss depth information, as shown in Fig. 4. The final reconstructed 3D model and camera trajectory are shown in Fig. 5, which demonstrates that our algorithm can handle this challenging case successfully.

V. Feature Matching

We define a reference feature set and initialize it by adding the extracted features on the first frame. Then for the following

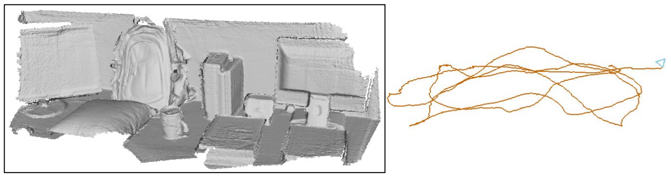


Fig. 5. The reconstruction result from the captured RGB-D data with significant depth missing. Left: the reconstructed 3D model. Right: the recovered camera trajectory.

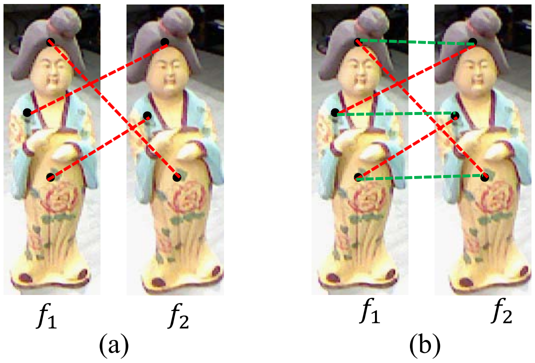


Fig. 6. Illustration of multi-candidates feature matching. (a) Features in frame f_1 may find false correspondences on frame f_2 through the nearest feature matching. The false correspondences are highlighted by red dotted lines. Using these correspondences, two frames will not be aligned. (b) Each feature in frame f_1 searches multiple nearest neighboring features on frame f_2 as correspondence candidates. The correct correspondences are included in the multi-candidates and highlighted using green dotted lines. By searching among all correspondences, two frames can be registered successfully.

frames, we match the extracted features with the reference ones, and estimate the camera pose with the proposed Prior-based Multi-candidates RANSAC. The correspondences found by feature matching may contain many outliers, so they cannot be directly used for bundle adjustment. Instead, more accurate and dense correspondences are found according to the 3D position distances between the registered features of the current frame and the reference features.

A. Multi-Candidates Feature Matching

The 2NN heuristic matching method described in [35] highly relies on the distinctiveness of the features, which may have problems in the challenging cases with repeated textures/structures. In addition, it is also very likely that the found correspondence with the most similar descriptor is not the correct one, especially in the case with repeated textures/structures. So it is not robust to use a standard matching method to find correct correspondences. Here, we propose to find multiple nearest neighboring features from the reference feature set as the correspondence candidates for each feature on the current frame. An illustration is shown in Fig. 6. The set of found neighboring features is denoted as $C = (c_1, c_2, \ldots, c_b)$ in ascending order of feature descriptor distance. It is reasonable to assume that the correct correspondence is in the candidate set, so the following procedure is to find the correct one.

In order to handle fast camera motion, for frame i, we first match the extracted features with those on frame $i - 1$, and estimate its camera pose. Then we transform the features of frame i to the reference 3D coordinate with the estimated camera pose for initial alignment, and search the correspondences from the neighboring reference features. The searching range is limited to a small sphere which is centered at the mapped feature. The radius of the sphere is generally set to $0.05m$. This local searching strategy not only can accelerate the matching process but also improve the robustness.

Algorithm 1 A Brief Summary of Our Proposed PMCSAC

```
1: Sample a minimal subset from the input data according to the inlier distribution information of previous frames defined in Equation (6) and Equation (8).
2: Estimate the model parameters to fit this minimal subset.
3: Evaluate the current model hypothesis.
4: If currently best hypothesis has occurred, refine the model parameters using all inliers of the best hypothesis and update the inlier ratio.
5: Repeat 1-4 until the probability of finding better hypothesis falls under predefined threshold, as in Equation (11).
```

B. Prior-Based Multi-Candidates RANSAC

It will be time-consuming to use a traditional RANSAC to find a good hypothesis if the inlier ratio is rather low. Here, we propose a Prior-based Multi-Candidates RANSAC (PMCSAC) to handle this problem. Inspired by [39], since both depth and color image contents are similar among adjacent frames, we can use the inlier distribution information from the previous frames to guide the point sampling of the current frame. The sample with higher inlier probability is more likely to be selected than a sample with a lower value. In this way, an all-inlier minimal subset can be retrieved much earlier. In addition, we fuse the idea of local optimization proposed in LoSAC [40] into our PMCSAC framework. The procedure of the PMCSAC algorithm is summarized in Algorithm 1. The PMCSAC algorithm can be extended for a variety of model estimation. In the case of pose estimation, the inputs of the algorithm are the matched feature correspondences between current frame and the reference feature set, and the inlier distribution information of previous frames. The outputs are the pose parameters of the current frame. Sampling strategy, hypothesis evaluation and termination criterion, will be discussed below. Different to [39], our sampling strategy is defined in 3D space, and supports multiple candidates.

1) Sampling Strategy: Generally, we evenly divide the whole space surrounding the object into $30 * 30 * 30$ grids. In each grid G_i, we record the inlier ratio of the previous frames. Assume that n features accumulate in grid G_i and each feature has b correspondence candidates. If n_c features find the correct correspondences among their candidates, we can estimate the probability of sampling an inlier in the grid G_i as

$$\varepsilon_i^* = \frac{n_c}{n}. \tag{5}$$

We then define the probability distribution as

$$p_i = \varepsilon_i^* / \sum_j \varepsilon_j^*, \tag{6}$$

where p_i denotes the probability of sampling a point from G_i. After sampling a point $\mathbf{x}_i$ from a grid, the correspondence from all b candidates is selected for $\mathbf{x}_i$. According to the 3D Euclidean distance between each candidate and the point, we define the probability for sampling candidate c_k as

$$q_k = \frac{\delta}{d+\delta}, \tag{7}$$

where d is the 3D Euclidean distance between candidate c_k and $\mathbf{x}_i$, and δ is a parameter for adjusting the effect of distance on the probability. The probability for sampling candidate c_k is normalized as

$$q_k^* = \frac{q_k}{\sum_{j=1}^{b} q_j}. \tag{8}$$

Thus, a point sample for the current frame is drawn in two steps. In the first step, a point is selected from a grid according to probability defined in Equation (6). In the second step, the correspondence candidate of the point is sampled according to candidate sampling probability defined in Equation (8).

2) Hypothesis Evaluation: Since the outlier number may be much larger than the inlier number, we do not use the inlier number as the evaluation criterion for a hypothesis. We evaluate a hypothesis using the distribution of inliers. The distribution of inliers is described by a covariance matrix,

$$C = \frac{1}{N-1}\sum_i (f_i - \bar{f})(f_i - \bar{f})^T, \tag{9}$$

where N is the number of inliers, f_i is the 3D position of ith inlier feature, and $\bar{f} = \frac{1}{N}\sum_i f_i$ is the mean 3D position of all inliers. The score of the evaluated hypothesis is calculated as

$$s = \frac{4\pi\sqrt{det(C)}}{3A}, \tag{10}$$

where A is the grid volume for normalization, and $\frac{4}{3}\pi\sqrt{det(C)}$ is the volume of the ellipsoid representing the inlier distribution. The intuition of this evaluation is that the best hypothesis should have a large number of inliers which are evenly distributed in the whole space.

3) Termination Criterion: We denote $\hat{\varepsilon}_i$ as the inlier ratio of G_i for the current best hypothesis. The probability of selecting an inlier sample is $\sum_i p_i\hat{\varepsilon}_i/b$. The probability of selecting an all-inlier minimal subset is $(\sum_i p_i\hat{\varepsilon}_i/b)^m$, where m is the size of minimal subset. For camera pose estimation in our case, $m = 3$. The number of the iterations K_s should satisfy the following termination criterion,

$$(1 - (\sum_i p_i\hat{\varepsilon}_i/b)^m)^{K_s} < \eta. \tag{11}$$

If the probability of all the K_s samples containing outliers is less than η, the sampling process will be terminated.

C. Building Correspondences With Refinement

With the matched 2D-3D and 3D-3D correspondences, we can employ the joint 2D+3D bundle adjustment framework proposed by [41] to jointly refine the camera motion and the 3D positions of the reference features. We divide the whole sequence into a set of subsequences $S = (S_1, S_2, \ldots, S_n)$ where each subsequence contains at most 30 frames. When we have recovered the camera poses of all frames in S_i, we will perform a local bundle adjustment for S_i (the 3D points and camera poses in previous subsequences $(S_1, S_2, \ldots, S_{i-1})$ are fixed). This strategy can refine the 3D positions of the reference features in time to alleviate the drift problem. For each frame $t \in S_i$, with the estimated camera pose, we can project its extracted features to the reference 3D coordinate, and search their nearest reference features according to the Euclidean distance of 3D positions. Taking into account of the potential drift problem, for each feature, we first search its closest correspondence from the refined reference features in the last subsequence S_{i-1}. If the distance is lower than $0.006m$, we think they are matched. Otherwise, we further seek its closest correspondence in S_i. This strategy can effectively reduce the false matching caused by drift problem.

VI. Global Refinement

Performing local bundle adjustment on consecutive subsequences can alleviate but not completely address the drift problem, especially for loopback sequences. We need to detect the loop closure, and build a loop closure constraint to eliminate the drift problem. According to the estimated camera poses, each frame can find its overlapping frames from the whole sequence. If two non-consecutive frames (i, j) have large overlapping, we will extract the reference features distributed in frame i and its neighboring frames. The reference feature subset is denoted as F_i. We also extract the reference features distributed in frame j and its neighboring frames. The reference feature subset is denoted as F_j. We align these two reference feature subsets. If more than 20% of feature points in F_i find correspondences in F_j, we think the loop closure is detected, and merge the matched reference features. With the updated reference features and correspondences, we can employ bundle adjustment to globally refine the camera motion and 3D points.

It will be time-consuming to perform bundle adjustment with a large number of frames and 3D points. Therefore, we propose to reduce the number of involved features, which can significantly speed up the bundle adjustment without reducing much accuracy. For each reference 3D feature g_i, we compute the frame set F_i where each frame has a feature corresponding to g_i. For each frame j in F_i, we count the number n_{ij} of feature correspondences with the reference feature set. If for all frames in F_i, $n_{ij} - 1$ is larger than a threshold κ, we will remove g_i and set $n_{ij} = n_{ij} - 1$. After the above process, the number of reference features can be significantly reduced but all frames still have at least κ corresponding reference features. This simple strategy works rather well in our experiments.

We investigate the effectiveness of error distribution using our global registration. The test RGB-D data (about 1,400 frames) is captured around a computer box. We compare our method with another strategy that assumes the camera returns to the original position and only matches the first and last frames. The Mean Correspondence Error (MCE)

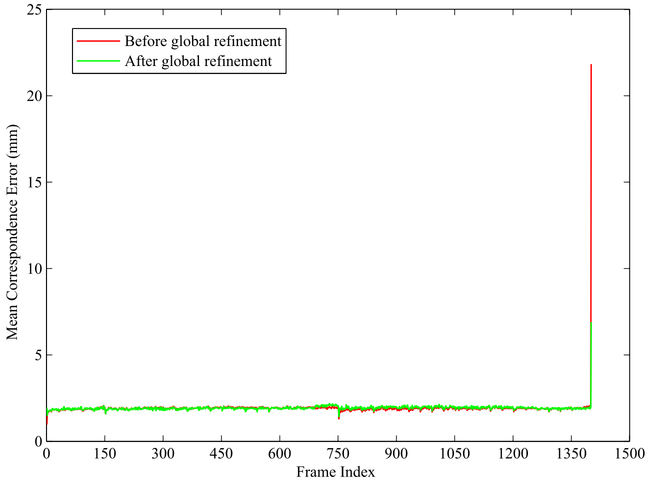


Fig. 7. Distribution of accumulation error using the simple loop closure constraint based on the correspondences between the first and last frames. The last frame has a large MCE error due to the additionally added correspondences with the first frame. After global refinement, the error is reduced but still much larger than the mean error.

for each frame measures the mean Euclidean distance of correspondences which is defined as

$$\mathbf{MCE}(i) = \frac{\sum_{k=1}^{m_i} \sqrt{d(\mathbf{Q}(T_i, g_j), p_{ik})^2}}{m_i}, \tag{12}$$

where p_{ik} is the kth 3D feature on frame i, g_j is the corresponding reference feature of p_{ik}, m_i is the number of feature correspondences with the reference feature set on frame i, and $\mathbf{Q}(T_i, g_j)$ is the transformed 3D position of reference feature g_j under transformation T_i. We compute the MCE for each frame before and after global refinement. Fig. 7 shows the error for each frame using the loop closure constraint based on the correspondences only between the first and last frames. We find that the last frame cannot align well with the first frame due to error accumulation. Before global refinement, the MCE error of the last frame is $21.8mm$. After global refinement, the MCE error is reduced to $6.9mm$ but still much larger than the mean error $1.9mm$. It demonstrates that using this simple loop closure constraint is still difficult to eliminate the drift problem. Fig. 8 shows the results of our method. Before global refinement, the last frames still have large MCE error. After global refinement, they are reduced to the mean error level, which demonstrates the effectiveness of our method.

Fig. 9 shows the reconstructed models of a computer box using different loop closure constraints. We find that the generated results are distorted due to large registration error using both Kinectfusion [5] and loop closure constraint between the first and last frames. The computer box includes two large planes, while Kinectfusion fails in the case when the sensor is faced by a large planar scene. In contrast, our method can recover the detailed model for the computer box with high registration accuracy. Note that in this example we only use the depth in a predefined bounding box surrounding the object in the reference coordinate frame.

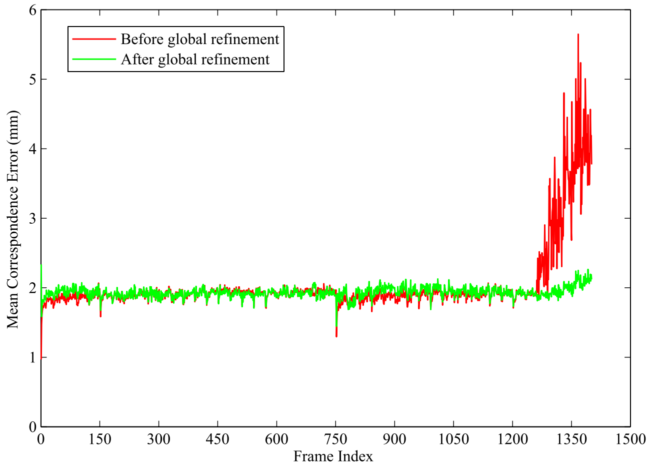


Fig. 8. Distribution of accumulation error using our method. The frames in the end part have high MCE error due to the correspondences with reference features distributed near the first frame. After global refinement, the error is successfully distributed through all frames evenly.

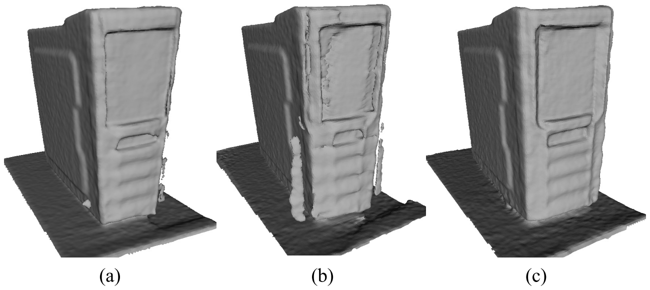


Fig. 9. The reconstructed computer box models using different methods. (a) Kinectfusion [5]. (b) Using the loop closure constraint only between the first and last frames. (c) Our method.

VII. Geometry Completion

To reconstruct a complete model for missing data, we first fuse all depth maps using TSDF and fill the missing depth for each frame through ray casting. Then, we recover the depth of each frame using the method proposed by [2], and further complete the missing depth of each frame using the proposed geometry completion algorithm.

Many factors, such as occlusion, textureless region, and strong reflection, can easily cause erroneous depth recovery by multi-view stereo (MVS) technique [2]. Generally, the scanned depth maps by Kinect are more precise than the recovered depth maps by MVS. By combining the captured depth data and recovered depth maps by MVS, we can complete the missing depth to obtain an as-complete-as-possible 3D model. For each frame, we triangulate the captured depth map to obtain a 3D mesh, and define it as target model. The 3D model generated with the recovered depth map by MVS is defined as source model. We use the surface registration method proposed in [42] to align the source model to the target model to complete the missing geometry. In [42], preliminary correspondences are found by a nearest point search and updated after each deformation. In our geometry completion, since the color images and captured depth maps are already calibrated, the correspondences between the source and target models can be easily determined and fixed. The surface registration method proposed in [42] defined three terms in its cost function, i.e. distance term, stiffness term, and landmark term. Based on the fixed correspondences, we only utilize distance term and stiffness term in the cost function of our geometry completion. Similar to [42], each vertex on the source model is assigned one affine $3 * 4$ transformation $\mathbf{X}_i$ that deforms the vertex into the corresponding vertex on the target model. The unknown transformations for all vertices are organized in a $4n * 3$ matrix, $\mathbf{X} := [\mathbf{X}_1 \cdots \mathbf{X}_n]^T$.

A. Distance Term

With the fixed correspondences $(\mathbf{v}_i, \mathbf{u}_i)$ between the source model and the target model, the sparse matrix $\mathbf{D}$ which maps the $4n * 3$ matrix of unknown $\mathbf{X}$ onto displaced vertices can be defined as,

$$\mathbf{D} := \begin{bmatrix} \mathbf{v}_1^T & & & \\ & \mathbf{v}_2^T & & \\ & & \ddots & \\ & & & \mathbf{v}_n^T \end{bmatrix}, \tag{13}$$

$\mathbf{v}_i$ is represented with homogeneous coordinate $\mathbf{v}_i = [x, y, z, 1]^T$. The corresponding vertices on the target model are arranged in a matrix $\mathbf{U} := [\mathbf{u}_1, \ldots, \mathbf{u}_n]^T$. The distance term is expressed under Frobenius norm $\| \cdot \|_F$,

$$E_d(\mathbf{X}) = \|\mathbf{W}(\mathbf{DX} - \mathbf{U})\|_F^2, \tag{14}$$

where $\mathbf{W}$ is a weight matrix denoted as $\mathbf{W} := \text{diag}(w_1, \ldots, w_n)$. If a vertex on the source model does not find the corresponding vertex on the target model, its weight w_i is set to zero.

B. Stiffness Term

The stiffness term is defined to penalize the differences between transformations of neighboring vertices using a weight matrix $\mathbf{G} := \text{diag}(1, 1, 1, \gamma)$.

$$E_s(\mathbf{X}) = \sum_{i,j \in \varepsilon} \|(\mathbf{X}_i - \mathbf{X}_j)\mathbf{G}\|_F^2. \tag{15}$$

In our case, the edge set ϵ is obtained according to neighboring pixels. If two pixels are neighbors, there is an edge between their corresponding vertices. The edges and vertices of the source model are numbered and its edges are directed from the lower numbered vertex to the higher numbered one. If edge r connects the vertices (i, j), the nonzero entries of the node-arc incidence matrix $\mathbf{M}$ in row r are $M_{ri} = -1$ and $M_{rj} = 1$. The stiffness term can be represented in matrix form as follows,

$$E_s(\mathbf{X}) = \|(\mathbf{M} \otimes \mathbf{G})\mathbf{X}\|_F^2, \tag{16}$$

where $\otimes$ is Kronecker product.

By combining the distance and stiffness terms, we define the following registration cost function similar to [42],

$$E(\mathbf{X}) = E_d(\mathbf{X}) + \alpha E_s(\mathbf{X}), \tag{17}$$

where α is the stiffness weight. The registration cost function can be rewritten in the matrix form,

$$\begin{aligned} E(\mathbf{X}) &= \left\| \begin{bmatrix} \alpha\mathbf{M} \otimes \mathbf{G} \\ \mathbf{WD} \end{bmatrix} \mathbf{X} - \begin{bmatrix} \mathbf{0} \\ \mathbf{WU} \end{bmatrix} \right\|_F^2 \\ &= \|\mathbf{AX} - \mathbf{B}\|_F^2 \end{aligned} \tag{18}$$

The cost function can be minimized and solved in the least-square sense.

In order to simultaneously keep the geometry detail of the source model and smoothly connect with the target model, the stiffness weight α is adjusted in a descending manner during deformation as [42]. In the beginning, α is set to a large value to rigidly align the source model to the target model. Then, α is set to a smaller value to allow more local deformation. When the change of $\mathbf{X}$ is lower than a threshold, the deformation converges. Then, we can use the corresponding deformed geometry to fill the missing parts on the target model. After completing depth maps of all frames, we can construct the complete 3D model using either Poisson surface reconstruction method [33] or the surface reconstruction method by KinectFusion [5].

VIII. Experiments

To evaluate the performance of the proposed approach, we have conducted various experiments (including human head and body modeling). All the RGB-D datasets are captured by a Kinect.

We measure the average computation time for each step of our algorithm on a desktop PC with Intel i5 3.1 GHz CPU and Nvidia GeForce GTX 560 SE display card. Table I lists the time statistics of different examples. We use parallel computation during feature extraction and matching process with four threads. The major computation is spent on bundle adjustment and global refinement. The times of bundle adjustment and global refinement, the total running time, and normalized time (seconds per frame) are listed for each dataset. Since serious depth missing occurs in the "Desk" and "Indoor" datasets, we need to use 2D-3D and 2D-2D correspondences for camera pose estimation, therefore the feature matching time is larger than other examples. In the "Synthetic Head", "Porcelain", and "Pottery" datasets, the frame number of each subsequence on which local bundle adjustment is applied, is set to 4, 5 and 6 respectively. In other datasets, it is set to 30.

A. Evaluation of Global Registration

We first compare our global registration with other two methods (i.e. Global-ICP and Kinectfusion) using synthetic data. Masuda et al. [15] registered the new frame using ICP with the merged point cloud from all previously processed frames, which we call as Global-ICP. A merged global point cloud is generated by merging points on each frame into the global coordinate frame in this method. Kinectfusion [5] fused all previous frames using a volumetric representation (TSDF) [43], rendered a dense surface prediction by ray-casting the TSDF into the camera pose of previous frame, and tracked the new frame by registering it against the

TABLE I

TIME STATISTICS FOR THE EXAMPLES USED IN OUR EXPERIMENTS. AVERAGE RUNNING TIME IS COMPUTED FOR EACH STEP: FEATURE EXTRACTION, FEATURE MATCHING, BUNDLE ADJUSTMENT, AND GLOBAL REFINEMENT. THE TOTAL RUNNING TIME OF THESE STEPS AND NORMALIZED TIME (SECONDS PER FRAME) ARE ALSO LISTED

Sequence	Frames	Average time				Total time	Normalized time
		FE	FM	BA (Times)	GR (Times)		
Synthetic Bear	134	0.070s	0.207s	0.521s (5)	7.461s (1)	49.715s	0.371s
Synthetic Head	14	0.070s	1.913s	0.334s (4)	3.480s (1)	32.842s	2.346s
Bag (Fig. 16)	680	1.124s	0.564s	2.484s (23)	79.639s (1)	21.491m	1.896s
Human Body (Fig. 17)	1930	0.263s	0.530s	6.785s (65)	50.690s (57)	1.399h	2.610s
Porcelain (Fig. 1)	109	0.224s	0.502s	0.685s (22)	98.684s (1)	3.246m	1.787s
Desk (Fig. 5)	1228	0.300s	1.060s	2.957s (41)	15.611s (32)	38.404m	1.876s
Computer Box (Fig. 9)	1400	0.267s	0.684s	8.919s (47)	63.923s (1)	30.467m	1.306s
Copy Room (Fig. 14)	5490	0.253s	0.636s	3.913s (183)	47.031s (165)	3.960h	2.597s
Indoor (Fig. 15)	8720	0.281s	1.103s	8.243s (291)	67.288s (272)	9.379h	3.872s
Pottery (Fig. 20)	120	0.243s	0.621s	0.330s (20)	81.178s (1)	3.207m	1.603s

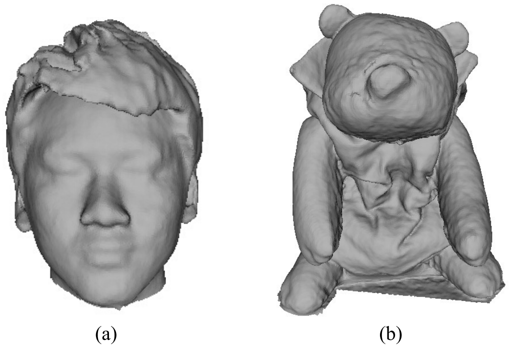


Fig. 10. Head and bear models for generating synthetic sequences. (a) Head model. (b) Bear model.

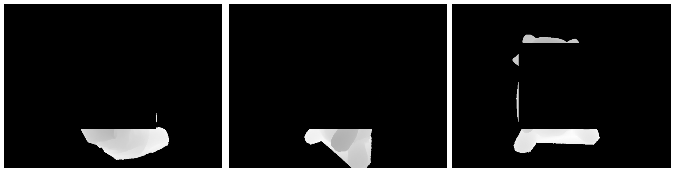


Fig. 11. The depth images of frames 66, 123, and 133 in the "Synthetic Bear" sequence. The black regions are those depth missing areas.

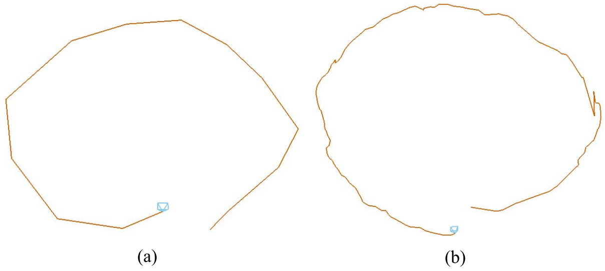


Fig. 12. (a) Camera trajectory of "Synthetic Head" data with 14 frames. (b) Camera trajectory of "Synthetic Bear" data with 134 frames.

surface prediction. To demonstrate the effectiveness of our global refinement, we also compare our result with that without global refinement.

Synthetic Data: We first generate a head model and a bear model using our proposed reconstruction method. The reconstructed models are shown in Fig. 10. We then produce the color images and the corresponding depth images of two synthetic sequences by rendering the head and bear models from surrounding viewpoints respectively. The "Synthetic Head" sequence consists of 14 frames with wide baselines. The "Synthetic Bear" sequence consists of 134 frames. Since the bear mainly lies in center of the rendered images, we set the depth values of all pixels within the 250 × 250 window centered at the depth images to zero. This makes sure more than half of the whole depth is lost on all frames. The depth of some frames are shown in Fig. 11. Fast camera motion and large missing depth makes the data challenging to be accurately registered. The camera trajectories of the two sequences are shown in Fig. 12. We make the resolution of rendered images nearly the same as the real data. Distances between synthetic camera and the model are always within $1.0m$. The noise level in depth data is about $2 \sim 3mm$ when Kinect data is captured at a range $0.8 \sim 1.0m$. So we add a random noise between $[-3mm, 3mm]$ into all rendered depth images to simulate the real data.

Since we know all ground truth camera parameters, the registration result can be evaluated easily. All points p ($p \in [1, \ldots, P]$) on frame i are transformed into the global coordinate frame using the estimated camera parameters and the ground truth parameters respectively, generating two transformed point clouds $X_e(p)$ and $X_g(p)$. The two transformed points should be identical if the parameters are accurately estimated. We compute the Root Mean Square error (RMSE) to measure the registration error by

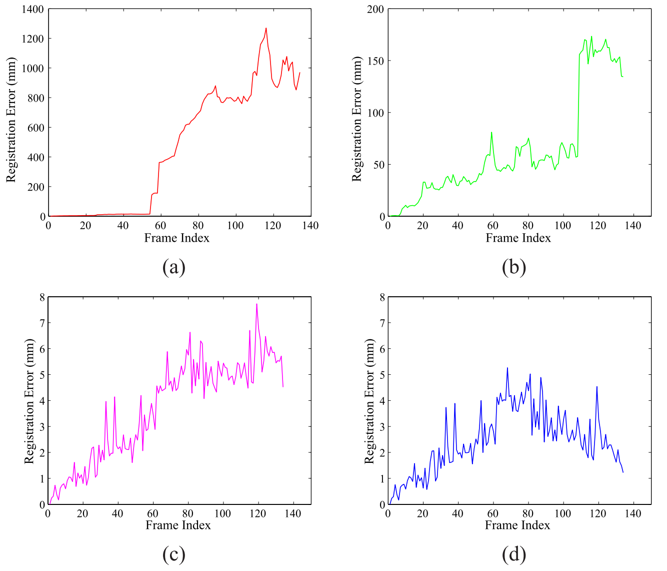
$$\mathbf{RMSE}(i) = \sqrt{\frac{1}{P}\sum_{p=1}^{P}\|X_e(p) - X_g(p)\|^2}. \tag{19}$$

Table II lists the measured registration error of each frame for "Synthetic Head" data and Fig. 13 shows the registration error of "Synthetic Bear" data. Tables III and IV show the average and maximum errors of different methods. The results show that our approach can align all depth images of each sequence with a lower error. In particular, the error without global refinement is higher than that with our global refinement. For "Synthetic Head" data, since the accumulation error is already rather small, the difference between with and without global refinement is also not obvious.

TABLE II

REGISTRATION ERROR(*mm*) FOR EACH FRAME OF "SYNTHETIC HEAD" DATA USING DIFFERENT REGISTRATION METHODS. SINCE THE FIRST FRAME IS SET TO THE REFERENCE FRAME, THE ERROR IS ZERO FOR ALL METHODS

Frame Index	2	3	4	5	6	7	8	9	10	11	12	13	14
Kinectfusion	112.608	86.450	90.226	134.805	213.469	214.720	205.526	204.606	191.240	175.256	572.332	716.902	559.589
Global-ICP	3.036	31.136	57.276	81.770	167.291	195.665	192.363	194.530	193.301	177.144	151.609	110.572	111.146
Without GR	1.171	1.840	2.085	3.087	4.901	4.329	3.906	2.478	2.274	2.368	2.470	4.775	2.785
With GR	1.168	1.838	2.073	3.053	4.884	4.274	3.842	2.413	2.254	2.313	2.422	4.811	2.892



(a)

(b)

(c)

(d)

Fig. 13. Registration error (*mm*) for each frame of "Synthetic Bear" data using different registration methods. (a) KinectFusion [5]. (b) Global-ICP [15]. (c) Our method without global refinement. (d) Our method with global refinement.

TABLE III

THE AVERAGE AND MAXIMUM REGISTRATION ERROR (*mm*) USING DIFFERENT METHODS FOR "SYNTHETIC HEAD" DATA

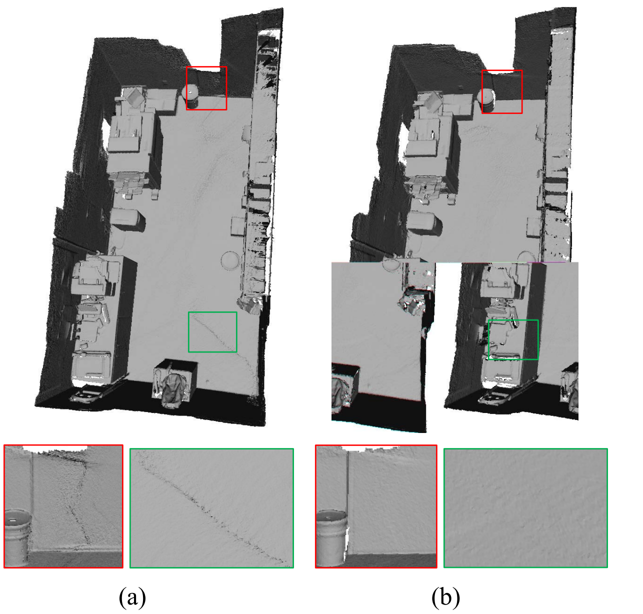
	Kinectfusion	Global-ICP	Without GR	With GR
Ave. error	248.409	119.060	2.748	2.731
Max. error	716.902	195.665	4.901	4.884

TABLE IV

THE AVERAGE AND MAXIMUM REGISTRATION ERROR (*mm*) USING DIFFERENT METHODS FOR "SYNTHETIC BEAR" DATA

	Kinectfusion	Global-ICP	Without GR	With GR
Ave. error	454.452	62.995	3.682	2.450
Max. error	1269.372	173.416	7.727	5.264

However, we find that fast camera motion and large missing depth dramatically affect the performance of Global-ICP and Kinectfusion. Independent registration error for each frame is fused into the global model, and the registration error accumulates. We see the error under Kinectfusion is higher than that of Global-ICP. This is because that Global-ICP aligns the new frame into the merged global point cloud directly while Kinectfusion aligns the new frame to the surface prediction generated from the previous frame pose. The registration error of previous frame affects the registration of the new frame.



(a)

(b)

Fig. 14. Comparison with [29]. (a) Zhou and Koltun [29]. (b) Our approach. Because the residual error is too large to be smoothed out, there are visible artifacts in the reconstructed surface by [29]. In contrast, our method can effectively eliminate the accumulated error and achieve better reconstructed surface.

We also capture a challenging indoor sequence with complex camera motion. Serious depth missing occurs in some frames. Fig. 15 illustrates the reconstruction result of the indoor scene. In addition, we extensively test the RGB-D data from [29]. The comparison result is shown in Fig. 14. In [29], relative pose estimates between connector frames are treated as soft constraints during global optimization of the pose graph. When the residual error is too large in relative pose estimates of some connector frames, the error cannot be eliminated completely. This leads to visible artifacts in the reconstructed surfaces, as shown in Fig. 14(a). In contrast, our method can achieve better result thanks to using bundle adjustment and global refinement based on point-to-point correspondences. The reconstructed 3D models of bag, bear, and human head are shown in Fig. 16. Fig. 17 shows the reconstructed 3D model of a human body.

B. Evaluation of PMCSAC

We compare the proposed PMCSAC algorithm with the standard RANSAC [44], LoSAC [40], and Multi-GS [45]. To test the effectiveness of the inlier ratio prior, we compare PMCSAC with MCSAC (without inlier ratio prior). The inlier

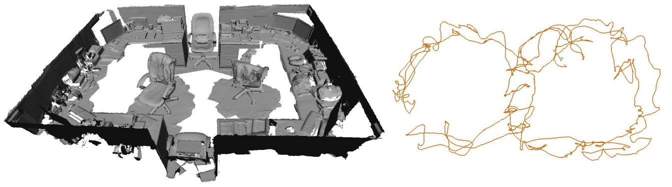


Fig. 15. Reconstruction result of an indoor scene. The length of camera trajectory (shown right) is about $73.5m$.

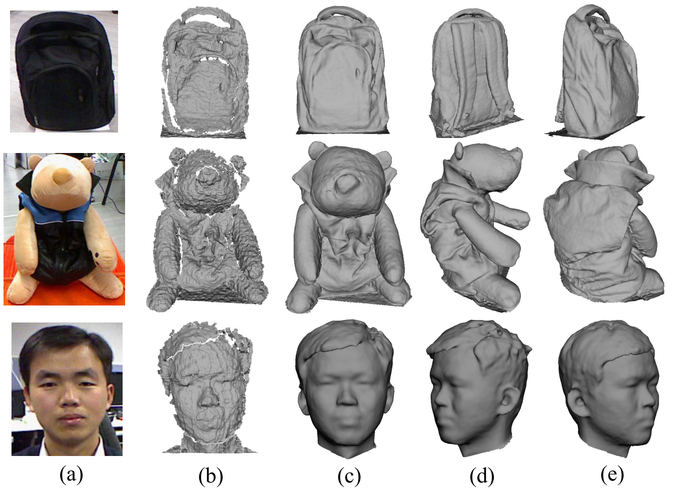


Fig. 16. Reconstructed 3D models by our method from bag, bear, and human head datasets, respectively. (a) A color image. (b) The corresponding captured depth data. (c)~(e) Different views of the reconstructed models.

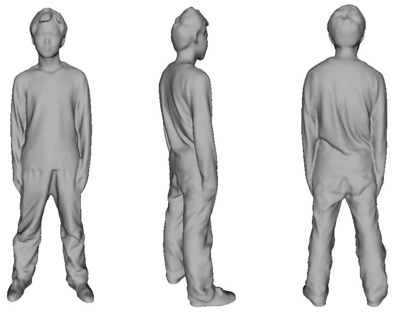


Fig. 17. The reconstructed human 3D model by our method.

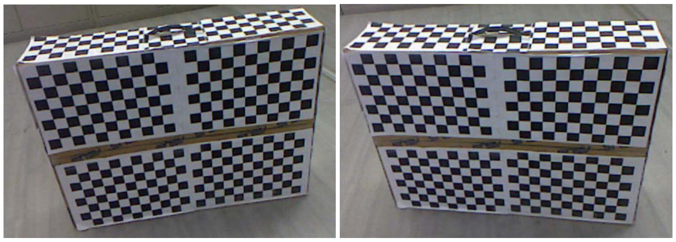


Fig. 18. The box data with checkerboard texture used in evaluation of PMCSAC.

TABLE V

THE RUNNING TIME OF DIFFERENT ALGORITHMS WITH LOCAL MATCHING

	RANSAC	LoSAC	Multi-GS	MCSAC	PMCSAC
Iterations	15,165	11,579	14,578	15,594	4,339
Ave. Time	2.141s	1.661s	906.887s	2.098s	0.622s
Total Time	0.611h	0.491h	226.797h	0.601h	0.231h

TABLE VI

THE RUNNING TIME OF DIFFERENT ALGORITHMS WITH GLOBAL MATCHING

	RANSAC	LoSAC	MCSAC	PMCSAC
Iterations	180,318	163,930	191,546	17,684
Ave. Time	29.944s	28.160	30.112s	2.967s
Total Time	7.558h	7.112h	7.600h	0.813h

threshold of LoSAC is set to $6mm$ as used in our method. The feature matching is performed with two strategies, local matching and global matching. The local matching strategy searches the corresponding features within a sphere around the feature with the radius of $50mm$, while the global matching strategy searches among all reference features. So the comparison is performed in the cases of local matching and global matching, respectively. We search 3 candidate correspondences from the reference features for each feature. For other methods, the nearest neighboring feature is found as the correspondence for each feature. We use all candidates of each feature for hypothesis evaluation. The termination criterion is the same for all algorithms as Equation (11), and the probability of selecting an inlier sample is computed according to the method of point sampling for different algorithms. We capture a sequence around a large box with checkerboard texture as the test data, as shown in Fig. 18. The sequence contains 900 frames in total. The average number of extracted features over all frames is $1,270$. The maximal iteration number is set to $20,000$ and $200,000$ for cases of local matching and global matching, respectively. Tables V and VI report the average running time and iteration number of all matching algorithms and total running time over the sequence. From the results, we can see that there is a remarkable improvement of our PMCSAC both in running time and iterations over other methods. Multi-GS requires a large running time due to high computational demands of sorting residual and computing preference correlation especially when the size of datasets is larger than 500. Under the case of global matching, the inlier ratio becomes much lower and more iterations are required. Because Multi-GS with global matching is too time-consuming (taking about $2,112s$ to process one frame), we do not make the experiment of Multi-GS with global matching.

We use the success ratio to qualify the convergence rate. The success ratio is computed as the score of current best hypothesis to that of the best hypothesis. The score of the best hypothesis is set to the maximal score obtained over all methods over all iterations. The success ratios of all methods along with the growing iteration number are shown in Fig. 19. From the results, we can see that our proposed PMCSAC can achieve a much higher success ratio at early stage of iterations and reach a stable state faster than other methods. Without inlier ratio prior, the performance of MCSAC is comparable with LoSAC. Multi-GS fails to estimate a good model at the

TABLE VII
THE RESULT STATISTICS USING DIFFERENT NUMBER OF CANDIDATES IN THE CASE OF LOCAL MATCHING

Candidate Number	1	2	3	4	5	6	7	8	9	10
Iterations	1,352	3,056	4,352	6,094	8,357	11,317	14,132	16,569	18,901	19,855
Running Time	0.133s	0.373s	0.591s	0.940s	1.417s	2.103s	2.903s	3.714s	4.678s	5.270s
Inlier Ratio	8.33%	14.59%	20.17%	24.92%	29.22%	32.92%	36.28%	39.32%	41.96%	44.36%

TABLE VIII
THE RESULT STATISTICS USING DIFFERENT NUMBER OF CANDIDATES IN THE CASE OF GLOBAL MATCHING

Candidate Number	1	2	3	4	5	6	7	8	9	10
Iterations	5,634	10,493	15,430	22,631	31,887	44,195	56,295	64,828	77,458	95,334
Running Time	0.568s	1.419s	2.408s	4.074s	6.447s	9.952s	13.897s	17.507s	23.463s	30.899s
Inlier Ratio	3.02%	5.00%	6.66%	8.10%	9.25%	10.33%	11.34%	12.28%	13.20%	13.97%

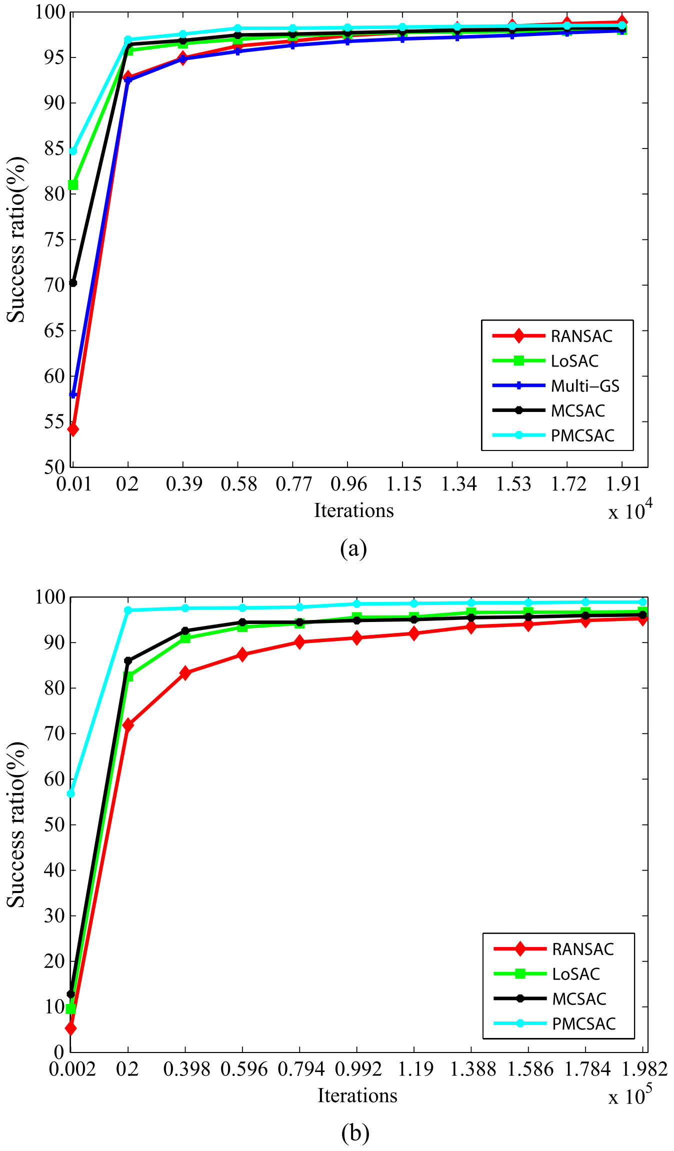


Fig. 19. The success ratio of different algorithms. (a) In the case of local matching. (b) In the case of global matching.

early stage since there are many outliers which make the preference analysis of Multi-GS unreliable. In contrast, our PMCSAC can achieve higher accuracy of model estimation in much less time.

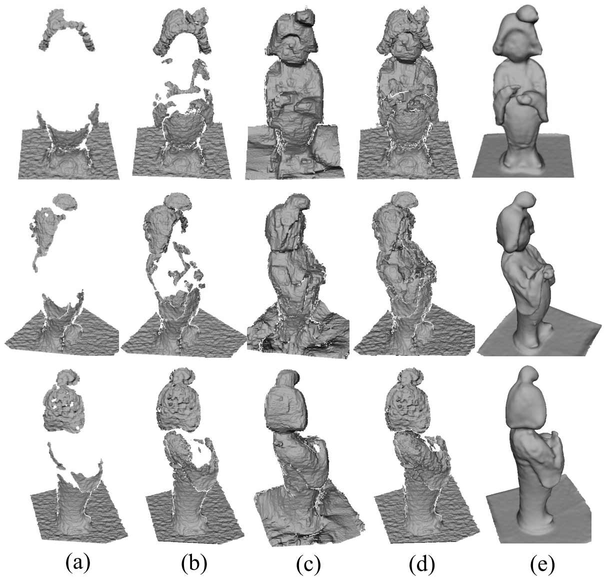


Fig. 20. Results of geometry completion. (a) Raw model captured by the depth sensor. (b) Completed model using captured depth from other frames. (c) Recovered model from the color image sequence. (d) Geometry completion result. (e) Different views of reconstructed model. Note that the missing parts in (a) are filled up.

We examine the effects of the number of candidate correspondences. The candidate number ranges from 1 to 10. For each number of candidate, we record the average running time, iteration number, and the inlier ratio. The inlier ratio is computed between the number of features that find correct correspondences and the total number of features. The results are shown in Tables VII and VIII. As the candidate number increases, the inlier ratio also becomes higher. However, the running time increases accordingly. Thus, it is better to strike a balance between the inlier ratio and the running time. In our experiments, we empirically use 1 $\sim$ 3 candidates.

C. Evaluation of Geometry Completion

In order to demonstrate the effectiveness of the proposed geometry completion algorithm, we capture a sequence around the pottery with two infrared flashlights interfering the depth sensor. To enlarge the camera motion, we only sparsely sample 120 frames. Influenced by the infrared, most depth maps miss over 50% depth information, so Kinectfusion [5] cannot handle this data. Please refer to the supplementary video for the complete frames. We first apply our global registration algorithm to register all frames into the same coordinate frame. We then fill up the missing depth of each frame with the depth information from other frames, as shown in Fig. 20(b). Then we use the color images to recover the depth maps for each frame by MVS technique [2], and use them to complete the depth data. Fig. 20(c) shows the recovered model by [2]. Although the geometry is relatively coarse and some parts are even inaccurate due to strong reflection and textureless problem, the major shape of the pottery is faithfully recovered. The geometry completion results are shown in Fig. 20(d). The final model is obtained by using the completed depth maps, as shown in Fig. 20(e). The geometry details are faithfully preserved, which demonstrates the effectiveness of the proposed method.

IX. Conclusion

In this paper, we propose a novel robust 3D reconstruction system with an RGB-D camera. We use visual and geometry features and combine SFM technique to make registration more robust especially in depth missing cases. In order to handle the repeated textures/structures, we propose a Prior-based Multi-Candidates RANSAC (PMCSAC) algorithm to make the feature matching more robust and efficient. We also use 3D information to help detect the loop closure and perform global refinement to eliminate the drift problem. The missing geometry due to depth missing can be effectively completed by combining multi-view stereo and mesh deformation techniques. Experimental results demonstrate that our method can achieve better 3D reconstruction results than the state-of-the-arts especially in challenging examples.

Currently, our system still cannot be applied to real-time applications. There is much room for our system to be accelerated since the proposed system is implemented using our unoptimized code. For long loopback sequences, the bundle adjustment and global refinement dominate the computation. There is already real-time bundle adjustment implementation [46] which can benefit our system. In addition, since our feature extraction and matching can be easily parallelized, we plan to use GPU to dramatically accelerate the system so that it can be applied in real-time applications in the future.

Acknowledgements

The authors would like to thank all the reviewers for their constructive comments to improve this paper. G. Zhang is the corresponding author.

References

[1] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. CVPR*, vol. 1. Jun. 2006, pp. 519–528.

[2] G. Zhang, J. Jia, T.-T. Wong, and H. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 974–988, Jun. 2009.

[3] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, "High resolution passive facial performance capture," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–4, 2010.

[4] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt, "3D shape scanning with a time-of-flight camera," in *Proc. CVPR*, Jun. 2010, pp. 1173–1180.

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

[6] T. Whelan, J. McDonald, M. Kaess, and J. J. Leonard, "Deformation-based loop closure for large scale dense RGB-D SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2013, pp. 548–555.

[7] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud, "Surface feature detection and description with applications to mesh matching," in *Proc. CVPR*, Jun. 2009, pp. 373–380.

[8] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool, "Hough transform and 3D SURF for robust three dimensional classification," in *Proc. ECCV*, Sep. 2010, pp. 589–602.

[9] S. Salti, A. Petrelli, F. Tombari, and L. D. Stefano, "On the affinity between 3D detectors and descriptors," in *Proc. 3DIMPVT*, Oct. 2012, pp. 424–431.

[10] F. Tombari, S. Salti, and L. D. Stefano, "Performance evaluation of 3D keypoint detectors," *Int. J. Comput. Vis.*, vol. 102, nos. 1–3, pp. 198–220, Mar. 2013.

[11] F. Tombari, S. Salti, and L. D. Stefano, "Unique signatures of histograms for local surface description," in *Proc. ECCV*, Sep. 2010, pp. 356–369.

[12] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1991, pp. 2724–2729.

[13] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proc. ACM SIGGRAPH*, Jul. 1994, pp. 311–318.

[14] Y. Pekelny and C. Gotsman, "Articulated object reconstruction and markerless motion capture from depth video," *Comput. Graph. Forum*, vol. 27, no. 2, pp. 399–408, 2008.

[15] T. Masuda, K. Sakaue, and N. Yokoya, "Registration and integration of multiple range images for 3D model construction," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 1996, pp. 879–883.

[16] K. Nishino and K. Ikeuchi, "Robust simultaneous registration of multiple range images," in *Proc. ACCV*, Jan. 2002, pp. 454–461.

[17] R. Benjemaa and F. Schmitt, "A solution for the registration of multiple 3D point sets using unit quaternions," in *Proc. ECCV*, Jun. 1998, pp. 34–50.

[18] S. Krishnan, P. Y. Lee, J. B. Moore, and S. Venkatasubramanian, "Global registration of multiple 3D point sets via optimization-on-a-manifold," in *Proc. Symp. Geometry Process.*, Jul. 2005, pp. 187–196.

[19] G. C. Sharp, S. W. Lee, and D. K. Wehe, "Multiview registration of 3D scenes by minimizing error between coordinate frames," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1037–1050, Aug. 2004.

[20] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auto. Robots*, vol. 4, no. 4, pp. 333–349, 1997.

[21] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3607–3613.

[22] F. Steinbrucker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Nov. 2011, pp. 719–722.

[23] M. Dou, L. Guan, J.-M. Frahm, and H. Fuchs, "Exploring high-level plane primitives for indoor 3D reconstruction with a hand-held RGB-D camera," in *Proc. Asian Conf. Comput. Vis.*, vol. 2. Nov. 2012, pp. 94–108.

[24] 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.

[25] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense RGB-D mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 5724–5731.

[26] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," in *Proc. RSS Workshop RGB-D, Adv. Reasoning Depth Cameras*, Jul. 2012.

[27] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments," *Int. J. Robot. Res.*, vol. 31, no. 5, pp. 647–663, Apr. 2012.

[28] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 1691–1696.

[29] Q.-Y. Zhou and V. Koltun, "Dense scene reconstruction with points of interest," *ACM Trans. Graph.*, vol. 32, no. 4, p. 112, 2013.

[30] N. Fioraio, W. Garage, K. Konolige, and W. Garage, "Realtime visual and point cloud slam," in *Proc. RGB-D Workshop Adv. Reasoning Depth Cameras Robot.*, Jun. 2011.

[31] D. Galvez-Lopez and J. D. Tardos, "Real-time loop detection with bags of binary words," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 51–58.

[32] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[33] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. Eurograph. Symp. Geometry Process.*, Jun. 2006, pp. 61–70.

[34] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vis. Conf.*, 1988, pp. 147–151.

[35] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[36] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.

[37] N. Bayramoğlu, J. Heikkilä, and M. Pietikäinen, "Combining textural and geometrical descriptors for scene recognition," in *Proc. ECCV*, Oct. 2012, pp. 32–41.

[38] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–770, Jun. 2004.

[39] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular SLAM in dynamic environments," in *Proc. Int. Symp. Mixed Augmented Reality*, Oct. 2013, pp. 209–218.

[40] O. Chum, J. Matas, and J. Kittler, "Locally optimized RANSAC," in *Proc. 25th DAGM Symp. Pattern Recognit.*, Sep. 2003, pp. 236–243.

[41] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using SfM and object labels," in *Proc. ICCV*, Dec. 2013, pp. 1625–1632.

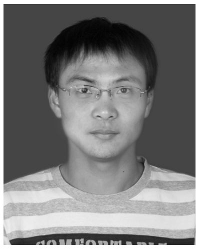
[42] B. Amberg, S. Romdhani, and T. Vetter, "Optimal step nonrigid ICP algorithms for surface registration," in *Proc. CVPR*, Jun. 2007, pp. 1–8.

[43] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. ACM SIGGRAPH*, Aug. 1996, pp. 303–312.

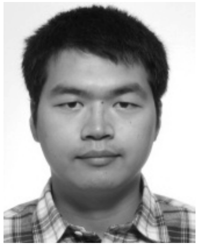
[44] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[45] T.-J. Chin, J. Yu, and D. Suter, "Accelerated hypothesis generation for multistructure data via preference analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 625–638, Apr. 2012.

[46] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *Proc. CVPR*, Jun. 2011, pp. 3057–3064.



Kangkan Wang received the B.S. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2009. He is currently pursuing the Ph.D. degree in computer science with the State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China. His research interests include face modeling, object tracking, and 3D reconstruction.



Guofeng Zhang (M'07) received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 2003 and 2009, respectively. He was a recipient of the National Excellent Doctoral Dissertation Award and the Excellent Doctoral Dissertation Award of the China Computer Federation. He is currently an Associate Professor with the State Key Laboratory of CAD&CG, Zhejiang University. His research interests include structure-from-motion, 3D reconstruction, augmented reality, and video segmentation and editing.

Hujun Bao (M'14) received the B.S. and Ph.D. degrees in applied mathematics from Zhejiang University, Hangzhou, China, in 1987 and 1993, respectively. He is currently a Cheung Kong Professor with the State Key Laboratory of CAD&CG, Zhejiang University. His main research interest is computer graphics and computer vision, including geometry and vision computing, real-time rendering, and mixed reality.