9-2013

A Robust RGBD Slam System for 3D Environment with Planar Surfaces

Po-Chang Su
University of Kentucky

Ju Shen
University of Dayton, jshen1@udayton.edu

Sen-ching S. Cheung
University of Kentucky

A ROBUST RGB-D SLAM SYSTEM FOR 3D ENVIRONMENT WITH PLANAR SURFACES

Po-Chang Su, Ju Shen, Sen-ching S. Cheung

Center for Visualization and Virtual Environments, University of Kentucky, Lexington, KY 40506

ABSTRACT

With the increasing popularity of RGB-depth (RGB-D) sensors such as the Microsoft Kinect, there have been much research on capturing and reconstructing 3D environments using a movable RGB-D sensor. The key process behind these kinds of simultaneous location and mapping (SLAM) systems is the iterative closest point or ICP algorithm, which is an iterative algorithm that can estimate the rigid movement of the camera based on the captured 3D point clouds. While ICP is a well-studied algorithm, it is problematic when it is used in scanning large planar regions such as wall surfaces in a room. The lack of depth variations on planar surfaces makes the global alignment an ill-conditioned problem. In this paper, we present a novel approach for registering 3D point clouds by combining both color and depth information. Instead of directly searching for point correspondences among 3D data, the proposed method first extracts features from the RGB images, and then back-projects the features to the 3D space to identify more reliable correspondences. These color correspondences form the initial input to the ICP procedure which then proceeds to refine the alignment. Experimental results show that our proposed approach can achieve better accuracy than existing SLAMs in reconstructing indoor environments with large planar surfaces.

Index Terms— 3D Reconstruction, Iterative Closest Point (ICP), Truncated Signed Distance Function, Ray casting TSDF, Large-scale planar surface alignment

1. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is a technique that uses a mobile camera to reconstruct an unknown 3D environment. Recent works such as [1] focus on using structured-light RGB-D cameras like the Microsoft Kinect to capture both the color and depth data by moving the RGB-D camera by hand through a large environment. Depth images captured by the moving camera are first projected onto a moving 3D coordinate system or a camera pose to create a cloud of 3D points. Based on the similarities between 3D point clouds captured at consecutive time instances, a rigid transformation is then estimated between the two camera poses. Such a process is performed over the entire sequence and a globally-consistent alignment of all point clouds can be obtained by repeated applications of the sequence of estimated rigid transformations. The global alignment is crucial for the final step of aggregating all the point cloud data into a volumetric representation for noise removal and rendering. The most commonly-used approach to estimate rigid transformations of camera poses from 3D point clouds is the iterated closest point or ICP algorithm [2, 3, 4]. The alignment accuracy of ICP significantly depends on the scene content. Figure 1 shows a virtual view of a vertical wall rendered from a 3D structure created by applying the ICP algorithm from [1] to align 50 frames of moving depth images. One can clearly see that the scene points are grossly misaligned.

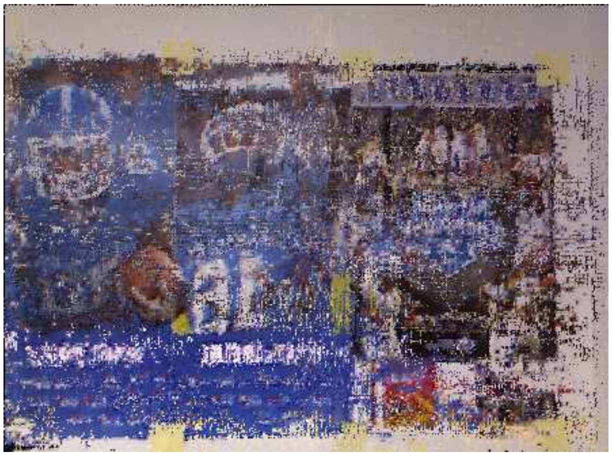


Fig. 1. Misalignment of a planar surface

The misalignment is caused by the failure of ICP in identifying correct correspondences between planar point clouds of successive frames. Such misalignment error accumulates over multiple frames, making it impossible to process a longer sequence. This is a significant shortcoming of ICP as vertical walls are common in indoor environments. In this paper, we propose a novel approach that can accurately reconstruct 3D indoor environments with large planar surfaces using both color and depth features. Our SLAM pipeline is based on that from [1]. An important difference is the use of color feature descriptors in improving depth data correspondences. Color feature descriptors are first identified from the color images and their correspondences across different frames are robustly identified. These correspondences are then projected onto the 3D coordinate system where they undergo a second stage of noise removal. An initial camera pose transformation is finally estimated which serves as the starting point of the iterative ICP process on the depth data. Our contribution is the development of this new joint color-depth alignment algorithm which produces significant better alignment than those from [1] as demonstrated by our experimental results.

The rest of this paper is organized as follows: we first define the problem in Section 2 and provide a survey on different SLAM systems in Section 3. In Section 4, we present the details of our proposed algorithm. Experimental results are shown in Section 5, followed by conclusions and future work in Section 7.

2. PROBLEM STATEMENT

To understand the problem on aligning planar surfaces, let us first review the basic procedure of ICP as summarized in Algorithm 1 [2]. Given two consecutive frames of 3D point clouds F_{t-1} and F_t, the

Algorithm 1 ICP

```
Require: $F_{t-1} = \{p_1, p_2, \dots, p_m\} \in \mathbb{R}^3$
         $F_t = \{q_1, q_2, \dots, q_n\} \in \mathbb{R}^3$
1: Initialization:
       $s := 0$ and $F^{(s)} := F_t$
2: Identify closest points: $\forall p_i \in F_{t-1}$
       $d_i := \min_{q \in F^{(s)}} \|p_i - q\|_2$
       $f^{(s)}(p_i) := \begin{cases} \arg\min_{q \in F^{(s)}} \|p_i - q\|_2, & d_i \leq \epsilon \\ \text{unmatched} & \text{otherwise} \end{cases}$
3: Find $R^{(s)}$ and $t^{(s)}$ to minimize the average of
       $\|p_i - (R^{(s)} \cdot f^{(s)}(p_i) + t^{(s)})\|_2^2$
   among all $p_i$'s with a matching $f^{(s)}(p_i)$
4: Refinement:
       $F^{(s+1)} := \{q'_i : q'_i := R^{(R)} \cdot q_i + t^{(s)}\}$
5: $s := s + 1$
6: Go back to step 2 until error in step 3 is below a threshold
```

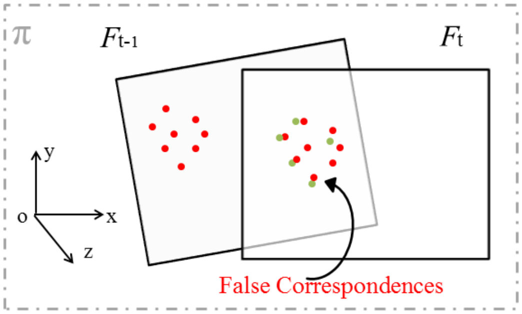


Fig. 2. How ICP fails to align planar structures

algorithm aims at iteratively refining a rotation matrix $R^{(s)}$ and a translational vector $t^{(s)}$ which are applied F_t to best align each point in F_{t-1} to its closest point in F_t. The distance parameter ϵ excludes correspondences that are too far apart to be considered as reasonable.

When ICP initially identifies the closest points between the two point clouds, there could be many false correspondences. The goal of ICP is to improve these correspondences by moving two point clouds closer to each other in each iteration. However, the above procedure may fail if the majority of the 3D points fall on a planar surface. This is illustrated in Figure 2. The red points from each plane indicate the true correspondences. But the closest-point search wrongly assigns the green points from F_{t-1} to match the points in F_t. If there were significant depth variations among the 3D points, no rigid transformation could produce a good match between these wrong correspondences and step 3 of the ICP algorithm merely produces a transformation that moves the two clouds closer. However, for a planar surface, these wrong correspondences may lead to a rotation about the x and the y-axis and a translation in the z direction that can completely align the two planes. The lack of depth variations prevents the in-plane rotation and the translation along the $x - y$ plane to be effectively estimated. As such, we have an underdetermined system and the ICP prematurely terminates without providing the true alignment. Notice that such misalignment error accumulates over time and thereby significantly affects the subsequent reconstruction of the 3D structure.

3. RELATED WORK

Among recent literature on SLAM systems, there have been a number of works on 3D environment reconstruction by using moving depth cameras [5, 6, 7, 1]. ICP algorithm is the most commonly-used technique to estimate the rigid transformations between consecutive frames by minimizing the overall correspondence points distances. However, using geometric information alone may suffer precision issues if the environment does not have enough spacial variations such as a large planar region.

To improve the alignment accuracy, a number of approaches have been recently proposed to combine color information together with geometric data [8, 7, 9]. In [8] and [10], color and depth data are integrated into a single weighted error function for alignment. However, color and depth offer different cues for alignment and a single error function with constant weights between color and depth is unlikely to work for different scenarios. In [11, 9], they propose similar schemes as ours by using features for initial estimation. But they use 3D points instead of voxels in the ICP procedure, which is prone to drifting error. To solve this problem, [9] uses TORO to optimize the estimation by globally recomputing the sequence of transformations between consecutive frames, which increases extra computational cost. The results are still less than unsatisfactory for planar regions, and additional postprocessing is required.

In contrast, in our proposed RGB-D mapping system, we perform color-feature based matching as part of the initial estimation, followed by the ICP procedure mapping each frame to a global TSDF-based voxel structure to further improve the registration accuracy. Based on the experiments, this combination yields desirable results.

4. PROPOSED APPROACH

Our SLAM system is based on [1]. A volumetric 3D grid structure called Truncated Signed Distance Function (TSDF) is used to aggregate 3D data obtained by depth images. Each voxel is signified by a signed distance value from the nearest 3D point. Using the reconstructed TSDF structure, virtual camera views can be rendered from an arbitrary camera pose – the virtual camera can ray-cast the TSDF structure to identify the estimative surface points. Specifically, the zero-crossing region is identified via a fast search procedure and the estimative surface points are interpolated within the zero-crossing region. During the model construction stage, ICP is used to estimate the camera pose with respect to the global coordinate system adopted by the TSDF structure. Instead of using the typical frame-to-frame tracking method which is prone to drifting error, the depth data are denoised and then aligned against the estimative surface points produced from the interpolated TSDF structure. Missing depth information is incrementally filled from depth data captured at different frames. Figure 3 is the overview of the SLAM system.

4.1. Joint color-depth camera pose estimation

The proposed camera pose estimation algorithm based on fused color and depth information works as follows. The RGB and depth cameras are extrinsically aligned and temporally synchronized using the OpenNI software library. Our camera pose estimation starts by first extracting the SIFT features from the color frames and search for the closest match between frames [12]. If all the SIFT feature points fall on the same plane, the matching correspondences between frames would be related by a planar homography. However, if the scene structure is more complex, we need a more robust procedure to identify the subset of correspondences that could fit well within a

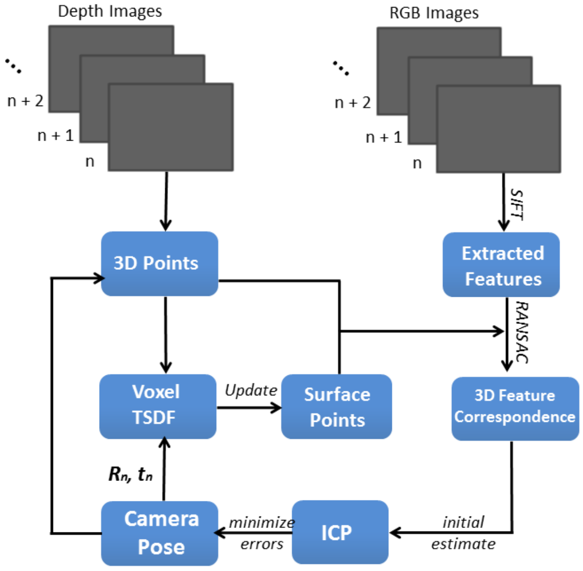


Fig. 3. Overview of the SLAM system

planar homography. To this end, we use a RANSAC-like procedure to identify such a subset and to estimate the optimal homography between correspondences [13]: all the correspondences are first used to estimate a homography matrix. We then apply the estimated homography to map one set of points to the other and eliminate those pairs that are too far apart. We repeat this process until it converges to a stable subset of correspondences that are well described by a single homography matrix. Finally these corresponding pairs along with the associated depth measurements are projected back onto the 3D space to be used as the initial point correspondences for the ICP algorithm.

Note that the SIFT correspondences are based on the current and previous color frames captured by the camera while the ICP is used to align the current depth frame with the TSDF voxel structure. Thus, we need to first raycast the TSDF structure using the estimated camera pose of the previous frame to recreate the point cloud from that frame. We then use the SIFT correspondences as the initial match between the recreated point cloud from the previous frame and the captured point cloud from the current frame. To ensure a robust matching, we again use RANSAC to find the inliers as a subroutine within ICP – outliers are iteratively removed if they do not agree with the estimated transformation until the procedure converges to a stable set of correspondences.

4.2. Virtual view rendering

After obtaining the estimative points by raycasting TSDF structure, each estimative point is mapped with a color value based on RGB image. In the raycasting process, each pixel has a corresponding ray passing through TSDF voxel structure, the pixel's color value is directly mapped to the corresponding estimative point. We use OpenGL to render the estimative points. Since the reconstructed scene is just a group of discretely distributed point cloud. The generated image may have many unrendered regions caused by the gaps between neighboring points. We apply a layered interpolation to fill in the gaps on the rendered image. This process can be described as follows: for the pixels with no display points, they need to be interpolated from neighboring pixels. A naive approach would be to perform spatial interpolation after obtaining the color values for all the pixels that contain at least one display points. We notice that this approach creates a great deal of blending of scene objects at different depth. To better preserve object boundaries, we separate the rendering into two phases based on the depth values from the scene points those that are at or closer than the viewer and those that are beyond. These two sets typically have very different depth values. We first start with the latter group with scene points that are far away, apply the above process of identifying color for each pixel and then perform interpolation on both depth and color values to fill in small gaps. These interpolated values are inserted back to the data structure of the closer pixels as if they are from the true 3D point clouds. In the second phase, we render all these closer pixels, select the correct color value based on both 3D point clouds and interpolated results, and finally perform one more round of interpolation just on the color values. Such a layered approach provides a far sharper object boundaries as it respects the inherent depth values. It is possible to increase the number of depth levels to create a better rendering but two levels are sufficient for our application.

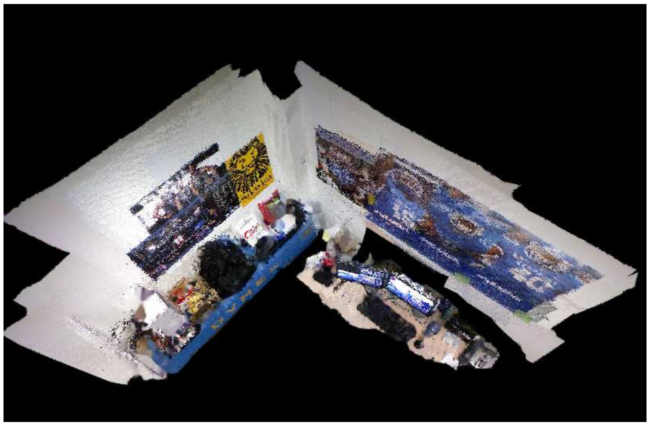


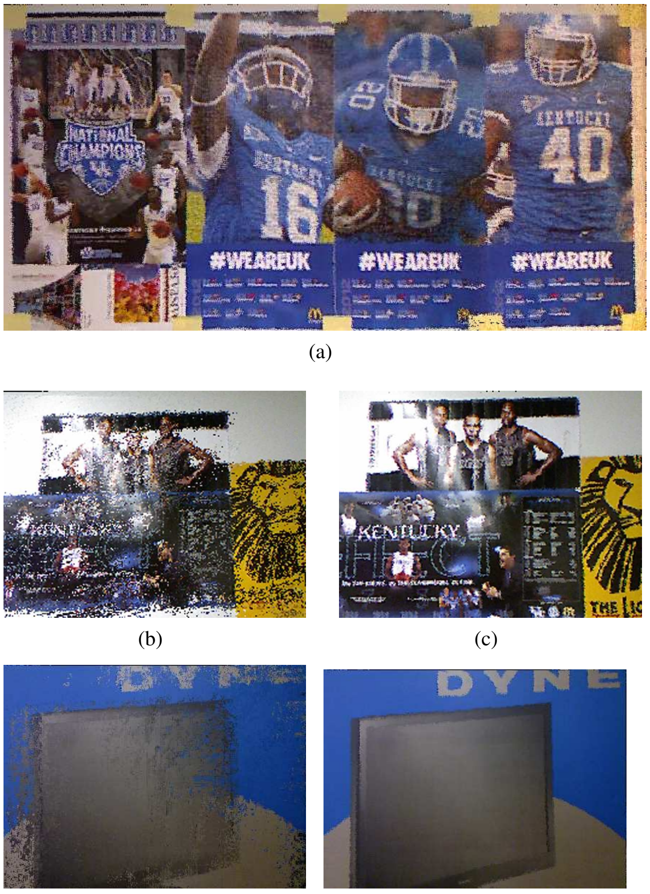
Fig. 4. Overview of reconstructed indoor environment

5. EXPERIMENTAL RESULTS

In our experiments, we test the proposed method by scanning a typical indoor environment with large planer surfaces. The experiment is conducted by using a single Microsoft Kinect. In Figure 4, it shows the 3D reconstruction result with the voxel size of $10 \times 10 \times 10$ mm. In terms of computation complexity, the only additional steps compared with [1] is the SIFT extraction and RANSAC matching, both of which can be run faster than real time. Using the parallel computation strategy as described in [14], our system can achieve real-time performance.

5.1. Qualitative Evaluation

To give a better analysis of the proposed method, we concentrate on the planar areas of the scanned environment and compare the reconstructed results with the ones by [1]. The reason for choosing [1] for comparison is that it represents a relatively popular approach that has been adopted by many other literatures [15, 16]. In Figure 5, the images are rendered by projecting the reconstructed 3D data to an arbitrary virtual view. Our results have significant improvements over the original scheme through a better preservation of the texture



(a)

(b)

(c)

(d)

(e)

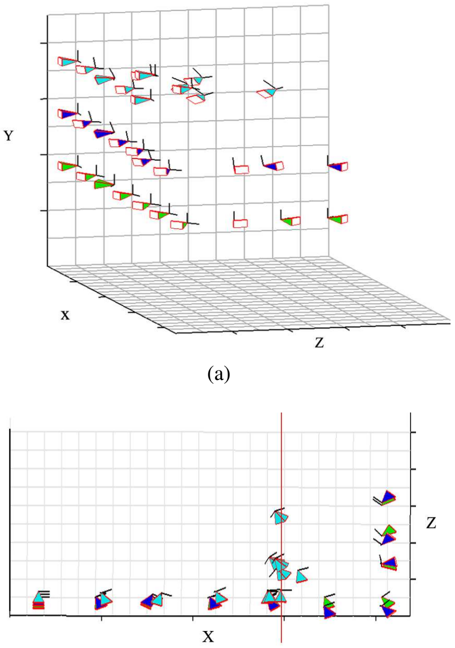
Fig. 5. Virtual views comparisons: (a) is our improved result with respect to the figure 1; (b)(d) and (c)(e) are the corresponding results by [1] and our method.

information on the planes. In particular, the text on the posters is clearly legible in the virtual views.

5.2. Quantitative Evaluation

For quantitative evaluation, we compare the estimated camera poses and locations against the ground truth, which is manually measured. We first use the Kinect to scan the environment against a predefined path. Along the path, we pick 10 arbitrary positions and physically measure the relative translation T and rotation R of the cameras on each spot. According to the manual measurements, a sequence of cameras are plotted in the 3D space as green cameras in Figure 6(a). Based on the associated frames on the spots, two sequences of transformations are estimated respectively by our proposed method and [1]. Their results are shown in the same figure. For demonstration purpose, we arbitrarily raise the blue cameras (our result) and cyan cameras ([1]) along the y axis by a fixed distance.

Figure 6(b) provides the top view of the results: the scan starts from the left side along the x axis and ends in the z direction. The total path is about $3.5m$. For the first few camera positions, all the three results are aligned closely due to the corresponding part of the captured environment involving considerable depth variation. After the red boundary, the camera enters large plane regions (the indoor wall), which causes the estimated cyan cameras to mess up. In contrast, our results are not affected by the planes and remain close to the ground truth.



(a)

(b)

Fig. 6. Camera pose estimation results: the ground truth is physically measured as the green cameras shows; the cyan cameras and blue cameras respectively indicate the results by [1] and our method.

Table 5.2 summarizes the estimation errors: the translation error T and rotation error R are statistically computed in terms of the offsets from the ground truth when the camera is scanned with a movement of 1.0m. The analysis is conducted by two different occasions depending on whether the scanned environment has dominant plane surfaces.

	Category	error T	error R
By [1]	nonplanar	$0.012m$	$0.208°$
	planar	$0.731m$	$39.20°$
Ours	nonplanar	$0.009m$	$0.224°$
	planar	$0.024m$	$0.875°$

6. ACKNOWLEDGEMENTS

Part of this material is based upon work supported by the National Science Foundation under Grant No. 1237134. Any opinions, ?ndings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily re?ect the views of the National Science Foundation.

7. CONCLUSIONS

In this paper, we have presented a novel point cloud registration approach that works well on planar surfaces. Color feature descriptors are first identified from the color images and their correspondences across different frames are robustly identified. These correspondences are then projected onto the 3D coordinate system where they undergo a second stage of noise removal. An initial camera pose transformation is finally estimated which serves as the starting point of the iterative ICP process on the depth data. All the depth data are aggregated in a voxel structure which is essential in reducing the drifting error and rendering virtual camera views.

8. REFERENCES

[1] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 127 – 136, 2011.

[2] J. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 239–256, Feb 1992.

[3] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," *International Conference on 3-D Digital Imaging and Modeling, 2001*, pp. 145 –152, 2001.

[4] H. Kjer and J. Wilm, "Evaluation of surface registration algorithms for pet motion correction," May 2010.

[5] R. Triebel and W. Burgard, "Improving simultaneous mapping and localization in 3d using global constraints," *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2005.

[6] P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schroter, L. Murphy, W. Churchill, D. Cole, and I. Reid, "Navigating, recognising and describing urban spaces with vision and laser," *International Journal of Robotics Research (IJRR)*, pp. 28(11–12), 2009.

[7] S. May, D. Droschel, E. Fuchs D. Holz, S. Malis, A. Nuchter, and J. Hertzberg, "Threedimensional mapping with time-of-flight cameras," *Journal of Field Robotics (JFR)*, pp. 26(11–12), 2009.

[8] L. Douadi, M. Aldon, and A. Crosnier, "Pair-wise registration of 3d/color data sets with icp," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 663– 668, 2006.

[9] 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," *International Journal of Robotics Research (IJRR)*, vol. 31, no. 5, pp. 647–663, April 2012.

[10] D. Neumann, F. Lugauer, S. Bauer, J. Wasza, and J. Hornegger, "Pair-wise registration of 3d/color data sets with icp," *IEEE International Conference on Computer Vision Workshops*, pp. 1161 – 1167, 2011.

[11] F. Endresand J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the rgb-d slam system," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, may 2012, pp. 1691 –1696.

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

[13] M. Fischler and R. 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, June 1981.

[14] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "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*. 2011, pp. 559–568, ACM.

[15] 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, Apr. 2012.

[16] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, and B. Guo, "An interactive approach to semantic modeling of indoor scenes with an rgbd camera," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 136:1–136:11, Nov. 2012.