2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Real-time 3D Reconstruction Using a Combination of Point-based and Volumetric Fusion

Zhengyu Xia, Joohee Kim
Illinois Institute of Technology, Chicago, US

Young Soo Park
Argonne National Laboratory

Abstract— **Real-time 3D reconstruction using low-cost commodity sensors like Kinect or Xtion has been successfully applied in a wide range of fields like augmented reality, robotic teleoperation, and medical diagnosis. Due to the assumption of static scene, popular 3D reconstruction technologies such as KinectFusion and KinFu, find truthful reconstruction with fast motion camera or segmenting a moving object to be a challenge. In this paper, we propose a weighted iterative closest point (ICP) algorithm that uses both depth and RGB information to enhance the stability of camera tracking. Additionally, a GPU-based region growing method that combines depth, normal and intensity level as similarity criteria, is also applied to segment foreground moving objects accurately. For real-time processing and GPU memory efficiency, we also design a combination of point-based and volumetric representation to reconstruct moving objects and static scene, respectively. Both qualitative and quantitative results show that our proposed method improves real-time 3D reconstruction on the performance of camera tracking and segmentation of moving objects with reduced computational complexity.**

I. Introduction

3D reconstruction like KinectFusion [1] or KinFu [2] has now become a well-established area of study in the field of robotics and computer vision. The objective is to recreate a real-world scene, such as an indoor room, in a three-dimensional virtual space with the help of geometric information of the said scene. Real-time 3D reconstruction has applications in areas like augmented reality (AR), robotic teleoperation, medical analysis, video games, etc. The availability of low-cost sensors like Microsoft Kinect [3] or Asus Xtion PRO and their quality of output and real-time properties have led to an increasingly more number of researchers and scientists taking an interest in real-time 3D reconstruction of physical scenes.

KinectFusion enables real-time 3D reconstruction of the static indoor environment using only depth information. It has five major processing steps as can be seen in Fig. 1:

1) Depth map refinement using bilateral filter [4].
2) Data measurement based on intrinsic sensor matrix.
3) ICP [5] algorithm to optimize pose estimation.
4) Surface reconstruction using TSDF [6] approach.
5) Raycasting [7] for surface prediction.

KinFu [2] is an open source implementation of KinectFusion and is a part of the Point Cloud Library (PCL). KinFu of PCL is used as the baseline software for this work.

The motivation for improvement in 3D reconstruction arises from the low-cost depth sensors mentioned earlier that have been shown to have systematic errors [8] in the depth, resulting in inaccurate camera pose estimation. Besides, one of the basic assumptions of KinectFusion and KinFu is that the scene under consideration is static, i.e., there is no scene motion independent of the camera. If the scene in question consists of movement independent of camera motion, it introduces tracking errors, and the 3D reconstruction of the scene gets distorted. In addition to this assumption, there are several other reasons which make it hard to segment and reconstruct moving objects. Firstly, ICP algorithm computes the optimal pose estimation, and thus any large movement would be considered as error and discarded. Secondly, volumetric representation used in KinectFusion for accurate model reconstruction has a complex data structure which requires a large amount of GPU memory. Hence, it is difficult to handle extra data like moving objects in real time.

To overcome the issues of systematic errors in depth, CI-ICP [9] combines the systematic error model with pose estimation process in the form of confidence indicator of each depth value. In CI-ICP, it is exploited that the pose can be estimated using objects closer to the camera. In ElasticFusion [10], a joint optimization of pose estimation which combines geometric and photometric information is proposed to deal with the errors in depth measurement and enhance the stable performance of camera tracking.

To efficiently handle the problems of segmentation and reconstruction on moving objects, the authors of point-based fusion [11] designed a system based on simple point-based representation during reconstruction, which works directly on the input acquired from the depth sensors. Apart from the point-based representation, this method is very much similar to that of the conventional 3D reconstruction methods discussed earlier. The advantage of point-based fusion [11] over volumetric representation is that the latter introduces computational complexity and memory overheads due to continuous conversions between different representations whereas the point-based fusion method works without the overhead of converting between representations. Along with that, it introduces the use of radius map [12] and confidence counter in the ICP algorithm to detect the dynamic candidates from non-corresponding points. These dynamic candidates are then segmented using a hierarchical region growing method and used for scene reconstruction. It further uses the surface splatting method [13] for the final 3D reconstruction.

Although [9] improves much on the camera pose, it still faces the challenge of estimating camera pose accurately when the captured depth maps are defected or contain holes due to the materials or the distance of objects. In [10], the photometric pose estimation is based on RGB information to find the minimum L2 distance between the current RGB pixels and the projected RGB pixels at previous frames. However, according to our observation and experiments, the

978-1-5386-8093-3/18/$31.00 ©2018 IEEE

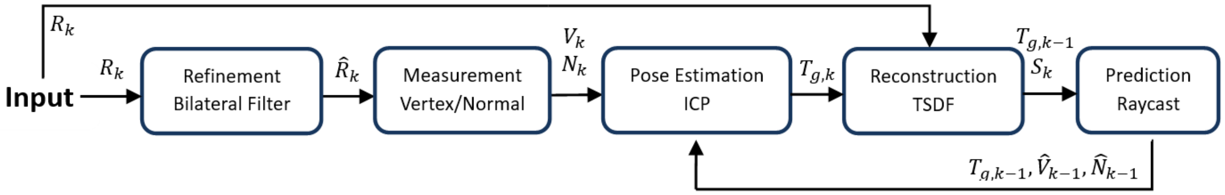

Figure 1. Processing Pipeline of KinectFusion/KinFu.

pose estimation in [10] highly relies on photometric optimization, which contributes 90% by default. It leads to poor performance when the camera motion is fast since the captured RGB maps contain a large number of inaccurate RGB pixels and the difference between current and previous RGB maps becomes larger resulting in unreliable alignments.

Though point-based fusion [11] solves the problems of computational complexity and memory overhead, it has some room to improve. First, the radius map and confidence counter mentioned above are one dimensional. However, as we know from the intrinsic parameters of the RGB-D sensor, it has separate focal lengths in x and y directions. The focal length of the depth sensor is required for radius calculation and using a single fixed focal length value may lead to inaccurate radius calculation and camera tracking. Secondly, the region growing method in [11] uses only depth and vertex map as the similarity attributes which may lead to the inclusion of static background as foreground moving parts. Thirdly, the reconstruction quality of the surface splatting method is not as good as the voxel-based reconstruction. Also, it has a time parameter to determine whether the current dynamic points should be added into the global model, which results in delay and increases the processing time.

To overcome these problems mentioned above in real-time 3D reconstruction of dynamic scenes, we propose a system that improves the camera tracking accuracy and increases the reconstruction quality of moving objects. Specifically, we propose a joint weighted ICP algorithm to obtain the optimal camera pose and adopt a combination of point-based and volumetric fusion to efficiently reconstruct moving objects and static background. To compensate for the systematic error in depth measurement, we assign a weight indicator for each depth value in geometric pose estimation. Considering the increasing error of pose estimation when the camera motion is fast, we replace L2 norm distance with L1 norm distance and decrease the weight ratio of photometric optimization. To tackle with the issues of point-based fusion [11], we extend the radius map and confidence counter to three dimensions and combine the point-based fusion and volumetric representation to maintain the level of the reconstruction quality while reducing the processing time and memory usage. We also include the normal and intensity map along with depth map as a similarity attribute to improve the accuracy of the region growing method and implement it on the GPU to maintain its processing speed. Then point-based reconstruction of the dynamic parts is combined with voxel-based reconstruction of the static background. The foreground moving objects are simultaneously reconstructed and added into the global model frame by frame. Hence it has no time-delay issue mentioned in [11]. Both quantitative and qualitative results show that our method outperforms other state-of-the-art methods such as [1], [2], [10] and [11].

The rest of this paper is structured as follows. Section II explains our proposed system for real-time 3D reconstruction. In Section III, the experimental results and analysis are given. Section IV concludes the paper along with a description of future work.

II. Proposed Method

A. Overall Architecture

In order to improve the performance of real-time 3D reconstruction of dynamic scenes, we propose a system which is inspired by the architectures of KinectFusion and the point-based fusion method [11]. For data measurement, we extend the radius map of the camera from one dimension to three dimensions. Each depth sensor has its own unique intrinsic camera matrix because of the difference in manufacturing process. Therefore, in the first stage of measurement, our model computes additional data - radius map $\boldsymbol{r}_k$ along with vertex map $\boldsymbol{V}_k$ and normal map $\boldsymbol{N}_k$ at the k^{th} frame as shown in Fig. 2. Meanwhile, since low-cost depth sensors have been shown to contain systematic errors in depth measurement, Confidence-Indicator based ICP (CI-ICP) [9] is used to incorporate these error models into geometric pose estimation. Considering that the error of photometric pose estimation increases when camera motion is fast, we adopt L1 norm distance instead of L2 norm distance and decrease the weight ratio of photometric optimization. Thereafter, an ICP labeling map is generated and updated for each frame to estimate the points of moving objects. Based on the results of classification of moving and static points, a morphological method is applied to remove stray points and noise. We then use a GPU-based region growing method to extract entire moving objects. Based on the efficiency of GPU memory, detected moving objects are reconstructed by using simple point-based fusion. 3D reconstruction of the static background is still based on volumetric representation in order to obtain high quality model and accurate prediction.

B. Data Measurement

In data measurement step, KinectFusion/KinFu transforms a refined raw depth map $\hat{R}_k$ into corresponding vertex map $\boldsymbol{V}_k$ and normal map $\boldsymbol{N}_k$ in camera space, which will be used to optimize 6DoF camera pose transformation $T_{g,k}$. Since one of our aims is to capture foreground moving candidates, we also introduce a radius map $\boldsymbol{r}_k$ into our proposed system at this stage.

Instead of setting the radius of each point cloud to a certain fixed value, it is estimated conservatively by covering one point cloud of the input data according to the theoretical accuracy limit of the sensor. The calculation for obtaining the radius associated with each vertex given in [12] is as follows:

$$r_k = \frac{1}{\sqrt{2}} \frac{\boldsymbol{V}_k^{\mathrm{g}}(z)}{\boldsymbol{N}_k^{\mathrm{g}}(z) f}, \tag{1}$$

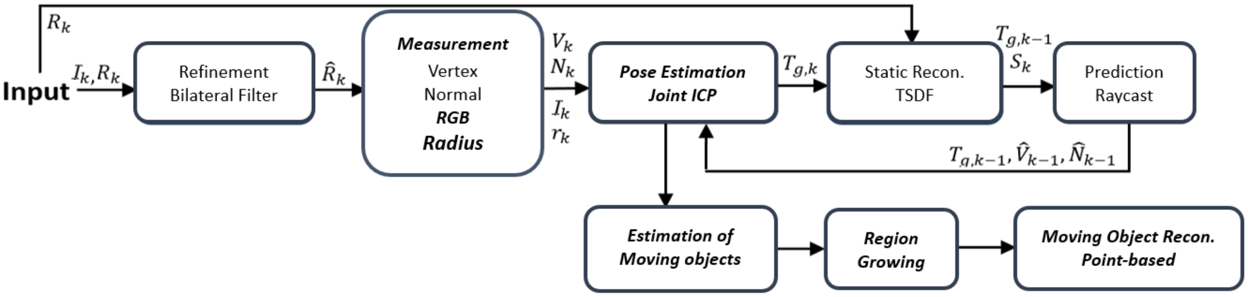

Figure 2. Proposed Processing Pipeline.

where f is the fixed focal length of the sensor, $\boldsymbol{V}_k^{\mathrm{g}}$ and $\boldsymbol{N}_k^{\mathrm{g}}$ are the global vertices and normal vectors computed using the current optimal transformation matrix $\boldsymbol{T}_{g,k}$, respectively. $\boldsymbol{V}_k^{\mathrm{g}}(z)$ is the depth value of the current global vertices while $\boldsymbol{N}_k^{\mathrm{g}}(z)$ is its corresponding global orientation along z direction.

Due to different manufacturing processes, each sensor has different focal lengths in x and y directions. Currently, various sensor calibration techniques such as [14] are designed to acquire more accurate intrinsic camera calibration matrix. Taking this into consideration, we extend the radius map to three dimensions. It helps calibrate the calculation of radius map accurately which will improve tracking ability and better selection of dynamic candidates. Consequently, the modified equation for the radius calculation is as follows:

$$\boldsymbol{r}_k = \begin{pmatrix} \boldsymbol{r}_k(x) \\ \boldsymbol{r}_k(y) \\ \boldsymbol{r}_k(z) \end{pmatrix} = \begin{pmatrix} \dfrac{1}{\sqrt{2}} \dfrac{\boldsymbol{V}_k^{\mathrm{g}}(z)}{\boldsymbol{N}_k^{\mathrm{g}}(z) f_x} \\ \dfrac{1}{\sqrt{2}} \dfrac{\boldsymbol{V}_k^{\mathrm{g}}(z)}{\boldsymbol{N}_k^{\mathrm{g}}(z) f_y} \\ \boldsymbol{V}_k^{\mathrm{g}}(z) \end{pmatrix}, \qquad (2)$$

where $\boldsymbol{r}_k \in R^3$, $\boldsymbol{r}_k(x)$ and $\boldsymbol{r}_k(y)$ are the radii of the point from calibrated focal lengths f_x and f_y point of view in x and y directions, respectively. $\boldsymbol{r}_k(z)$ is the distance from the camera center.

The radius map is used to compute whether the status of each point is stable or unstable by the following equation:

$$\boldsymbol{c}_k = \boldsymbol{c}_{k-1} + \boldsymbol{\alpha}, \qquad (3)$$

where $\boldsymbol{c}_k$ is a confidence counter that determines whether its point cloud is stable at the k^{th} frame. In our experiments, we have set the value of stability threshold to 10. That is to say, if the value of $\boldsymbol{c}_k$ at the point under consideration is larger or equal to 10, its status is considered as stable. $\boldsymbol{\alpha} = \exp\left(-\overline{\boldsymbol{r}_k}^2\right)$ is a confidence parameter, which applies a Gaussian weight on the current normalized radius map $\overline{\boldsymbol{r}_k}$.

C. Pose Estimation

The low-cost depth sensors face the challenge to accurately obtain camera pose when the captured depth maps are defected or contain holes due to the materials or the distance of objects. In [10], a photometric pose estimation based on RGB inputs is proposed. It finds the minimum L2 distance between the current RGB pixels and their projected RGB pixels at previous frames. However, in pose estimation [10], the ratio is highly biased towards photometric estimation which contributes as much as 90% to that of geometric estimation by default. Hence, it suffers from inaccurate pose estimation when camera motion is fast since the captured RGB are more likely to introduce inaccurate pixels and results in unreliable alignments.

To overcome the shortcomings mentioned above, we propose a weighted joint ICP method which indicates how much of photometric estimation is to be shared. We also assign a confidence indicator of each depth value in geometric pose estimation to reduce systematic error in the depth measurement. The proposed geometric pose estimation is given in Eq. (3).

$$E_D(\boldsymbol{T}_{g,k}) = \sum_{\substack{match \\ points}} \left\| w_D \left(\boldsymbol{T}_{g,k}\boldsymbol{V}_k - \boldsymbol{V}_{k-1}^{\mathrm{g}}\right)^T \boldsymbol{N}_{k-1}^{\mathrm{g}} \right\|_2, \qquad (3)$$

where $\boldsymbol{T}_{g,k}$ is the estimated camera pose at the k^{th} frame. $\boldsymbol{V}_k$ is the vertex map in camera coordinate at the k^{th} frame. $\boldsymbol{V}_{k-1}^{\mathrm{g}}$ and $\boldsymbol{N}_{k-1}^{\mathrm{g}}$ are the vertex map and normal map in global coordinate at the $(k-1)^{th}$ frame, respectively. w_D is the weight of measured depth value which is calculated as inversely proportional to the square of measured distance, as given in Eq. (4).

$$w_D = \left[\frac{\dfrac{1}{R_k^2} - \dfrac{1}{d_{max}^2}}{\dfrac{1}{d_{min}^2} - \dfrac{1}{d_{max}^2}} \right], \qquad (4)$$

where R_k is the raw depth value at the k^{th} frame. d_{max} and d_{min} are the maximum and minimum distance measured by the sensor, respectively.

Considering the increasing error of pose estimation when the motion of the camera is fast, L2 norm distance is replaced by L1 norm distance. In addition to these steps, we decrease the weight ratio of photometric pose estimation in the proposed joint ICP algorithm to diminish the errors in photometric estimation, which is given in Eq. (5).

$$E_{RGB}(\boldsymbol{T}_{g,k}) = \sum_{\substack{match \\ points}} \left| I_k\left(\left\lfloor \pi\left(\boldsymbol{K}\boldsymbol{T}_{g,k}^{-1}\boldsymbol{u}\right)\right\rfloor\right) - I_{k-1}(\boldsymbol{u}) \right|, \qquad (5)$$

where I_k is the image intensity map at the k^{th} frame and $\boldsymbol{u} = (\mathrm{u}, \mathrm{v})^T$ is the image coordinate. $\left\lfloor \pi\left(\boldsymbol{K}\boldsymbol{T}_{g,k}^{-1}\boldsymbol{u}\right)\right\rfloor$ performs the perspective projection and computes the corresponding projective location of $\boldsymbol{u}$. $\boldsymbol{K}$ is a matrix containing the intrinsic camera parameters.

Finally, the joint cost function is given as in Eq.(6).

$$E(\boldsymbol{T}_{g,k}) = E_D(\boldsymbol{T}_{g,k}) + w_{RGB}E_{RGB}(\boldsymbol{T}_{g,k}), \qquad (6)$$

where $w_{RGB} = 10\%$ is used in our experiments. To minimize this cost function, the Gaussian-Newton non-linear least-square algorithm is applied to obtain the optimal $\boldsymbol{T}_{g,k}$ at every frame.

D. Estimation of Moving Objects

In KinectFusion and KinFu, when a moving object is introduced into the current view of the sensor or a static object starts to move, tracking errors occur during ICP process. The points of moving object in the current frame will be considered as outliers, and they would not take part in the optimal computation of camera pose. As a result, it fails to capture the moving object in the reconstruction. Thus, failure of ICP is a strong indication that there is a moving object in the current scene. According to our observation, a moving object is usually surrounded by static background, which will be classified as corresponding stable points in ICP. Therefore, these static points close to the boundary of moving object can be used as a feature to classify the boundary of moving object. As a result, we create a gray-scale ICP labeling map that stores this information for each depth sample during the data association and pose estimation step. The ICP labeling map ICP_k stores one of the following values for each point:

a) invalid (black): no corresponding point found and no stable model points in its proximity.

b) dynamic (gray): no corresponding point found but stable model points in its proximity.

c) static (white): stable model correspondence found for the input depth sample.

Algorithm 1: Estimation of Moving Objects

```
Inputs: $\boldsymbol{V}_{k-1}^{g}, \boldsymbol{V}_{k}^{g}, \boldsymbol{N}_{k-1}^{g}, \boldsymbol{N}_{k}^{g}$ and $\boldsymbol{c}_k$
Outputs: $ICP_k$
// Check Invalid or Missing Points
  if $\boldsymbol{V}_{k}^{g}(z)$ is invalid
    then $I_k \leftarrow$ invalid
// Search Corresponding Points
  // fail to find corresponding points
  if $\|\boldsymbol{V}_{k-1}^{g} - \boldsymbol{V}_{k}^{g}\| > T_{dist}$ And $\|cross(\boldsymbol{N}_{k-1}^{g}, \boldsymbol{N}_{k}^{g})\| > T_{angle}$
    // check surrounding pixels' confidence counter values
    if count $(\boldsymbol{c}_k \geq c_{stable}) < T_{\#stable}$
    then $ICP_k \leftarrow$ dynamic
    else $ICP_k \leftarrow$ invalid
  // find out corresponding points
  else $ICP_k \leftarrow$ static
```

Algorithm 1 illustrates the details of estimation on moving objects. At first, $\boldsymbol{V}_{k}^{g}(z)$ will be checked if it contains valid depth value. If it is valid, point is passed on to the second step which evaluates whether it has a corresponding stable point. If the point is deemed to be a non-corresponding point based on T_{dist} and T_{angle}, which are the thresholds of distance and angle under standard ICP criteria, respectively, we compute the confidence or stability of the surrounding points as per Eq. (3). The surrounding size is defined as 3x3 for our experiment. In a case where the point itself is found to be a non-corresponding point and if most of its surrounding points are stable, then it is marked as a dynamic candidate. $T_{\#stable}$ is the number of surrounding stable points depending on user's choice.

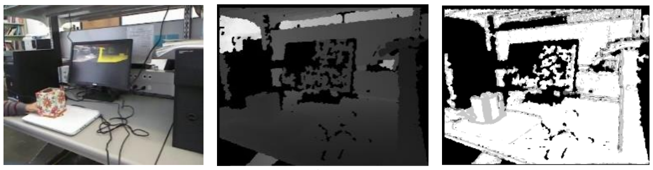
(a) (b) (c)

Figure 3. Generation of ICP labeling map for "Moving_box" dataset: (a) RGB image, (b) depth map, and (c) ICP labeling map.

Fig. 3 shows one example of testing scenes, where the moving object is a tissue box. After the estimation of moving object, a grayscale ICP labeling map is generated as shown in Fig. 3(c). The gray points indicate the candidates are the points of moving object. We observe that along with the tissue box; it has incorrectly labeled some of the background points in grey. In order to improve robustness to camera noise and occlusions, a GPU-based morphological erosion is applied to ensure that it only contains the inner region of dynamic objects. The next step is to find connected components in depth map R_k. Instead of performing hierarchical region growing [11], we propose a GPU-based region growing method which takes into account orientation and intensity differences between two points along with depth difference. Starting from these seed points, we collect similar points whose vertex $\boldsymbol{V}_k$, normal $\boldsymbol{N}_k$ and intensity I_k meet any two of the following conditions:

$$\left|\boldsymbol{V}_k^{seed}(\mathrm{z}) - \boldsymbol{V}_k^{neighbor}(\mathrm{z})\right| \leq T_{depth}, \qquad (7)$$

$$norm\left(cross\left(\boldsymbol{N}_k^{seed}, \boldsymbol{N}_k^{neighbor}\right)\right) \leq T_{orientiaon}, \qquad (8)$$

$$\left|I_k^{seed}(\mathbf{u}) - I_k^{neighbor}(\mathbf{u})\right| \leq T_{Intensity}, \qquad (9)$$

where $\boldsymbol{V}_k^{seed}(\mathrm{z})$, $\boldsymbol{N}_k^{seed}$ and $I_k^{seed}(\mathbf{u})$ are depth values, normal vectors and intensity level of the seed points, respectively. $\boldsymbol{V}_k^{neighbor}(\mathrm{z})$, $\boldsymbol{N}_k^{neighbor}$ and $I_k^{neighbor}(\mathbf{u})$ are depth values, normal vectors and intensity level of the points adjacent to the seed points, respectively. $cross$ calculates the angle between two normal vectors while $norm$ is the Euclidean norm of the vector. T_{depth} , $T_{orientiaon}$ and $T_{Intensity}$ are the thresholds of depth, orientation and intensity difference, respectively. In our experiments, we set the value of T_{depth} to 3 centimetres, $T_{orientiaon}$ to 20° and $T_{Intensity}$ to 3 pixel-level.

E. 3D Reconstruction using Combined Representation

Voxel-based fusion can reconstruct high-quality 3D model, but it also requires a huge amount of GPU memory. For a predefined volume (3m x 3m x 3m) with 512^3 resolutions, around 700 MB GPU memory would be occupied no matter how many voxels are used. Moreover, additional volume requires not only extra GPU memory but it also slows down the entire processing speed. In contrast, the GPU memory usage of point-based fusion only depends on how many point clouds you have in the current frame. According to our current experimental results, point-based fusion can achieve around 37.8 fps while dynamics estimation using volumetric representation can only achieve 15.8 fps.

In order to achieve high-quality 3D reconstruction with fast runtime, we use volumetric representation for static background while point-based fusion is used for foreground moving objects. The reasons are:

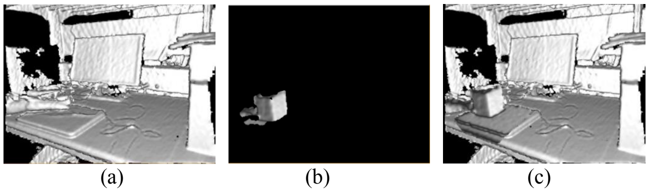

Figure 4. 3D Reconstruction for “Moving_box” dataset: (a) 3D voxel-based reconstruction of static background, (b) 3D reconstruction of moving object using point-based fusion, and (c) 3D reconstruction of the scene using combined representation.

1) Volumetric representation can achieve high-quality 3D reconstruction.

2) Volumetric representation applies TSDF along with Raycasting. Compared to point-based fusion, it estimates global vertices and normals more accurately, and accurate data prediction improves the camera tracking ability for the future frames. It also helps to detect the candidates of moving objects more precisely since it has better performance on ICP.

3) Point-based fusion can achieve faster processing speed using less GPU memory at the expense of reconstruction quality.

Once the parts of moving objects are estimated and computed, they are added using point-based fusion into the current static volume-based background. Any voxel in the static background is replaced by point-based representation if its current location is marked as dynamic. This step also helps eliminate the incorrect reconstruction of static background where it has motion as shown in Fig. 4(a). In addition, our proposed method achieves processing speed of 32.4 fps. Although its processing speed is lower than [11], it has more stable tracking ability and more accurate reconstruction of the foreground moving objects which are shown and discussed in Section III.

III. Experimental Results

A. Experimental Setup

It is difficult to make a meaningful qualitative and quantitative comparison of two algorithms using output of Kinect sensor directly because for each observation of the same scene we will get different sensed value. To circumvent this problem and to achieve an accurate comparison process, we created a framework to get quantitative data for comparing algorithms. We use the Munchen Technology University (TUM) RGB-D SLAM dataset which is a public-domain dataset with ground-truths used for the evaluation of visual odometry and visual SLAM systems [15]. The resolution of the depth map and the color image is 640 x 480. Since the ground truths of moving objects in TUM RGB-D Dataset are not provided, we adopt SBM-RGBD Dataset [16] to evaluate the performance of segmentation on foreground moving objects. Experiments were conducted on a workstation having six processor cores running at 3.2 GHz with 16 GB RAM and GTX 1080 with 8 GB GPU RAM as Graphics.

For the performance evaluation of camera tracking, two error metrics are used: Absolute Trajectory Error (ATE) gives the root mean square measure of error between the ground truth and the pose estimation. Relative Pose Error (RPE) measures the local accuracy of the trajectory over a fixed time interval Δ. The improved point-averaging method is based on point-based fusion, but the radius maps and confidence counters have been extended into three dimensions. This, along with the weighted joint ICP algorithm, and combined representation is our proposed method in the paper.

B. Tracking Ability

Since KinFu and ElasticFusion cannot deal with the scenes with moving objects, the experiments on the performance of camera tracking ability are conducted under the environments of static 3D reconstruction.

The point averaging method [11] can detect moving objects, yet it decreases the performance of tracking ability. As shown in Table I, it is found to perform worse than the baseline KinFu by 4.77% in ATE metric terms. Our proposed solution on the point averaging method uses calibrated focal length and extends the radius maps into three dimensions. As a result, the proposed processing pipeline with the improved point averaging method performs better than the baseline KinFu by 2.16% in ATE metric terms and 9.42% in RPE metric terms.

CI-ICP is effective in cases where objects with structure are present close to the camera. Photometric method based on L1 norm distance helps increase the accuracy of camera pose when depth maps are defected or contain holes. Compared with L2 norm distance measurement, it decreases the errors in pixel alignment from color information when camera motion is fast. In the proposed system, both methods are combined to get a robust 3D reconstruction algorithm. The experimental results given in Table I shows that our proposed method performs better than ElasticFusion [10] in fast motion dataset such as “Teddy” or “Plant”, while it is comparable to ElasticFusion in slow motion scenarios such as dataset “XYZ-1” or dataset “XYZ-2”. The proposed pipeline with the weighted joint ICP method and improved point averaging method outperforms the other state-of-the-art methods. Specifically, it performs better than the baseline KinFu by 28.15% in ATE and by 27.11% in RPE metric terms.

We also present the qualitative comparison of 3D reconstruction among three different methods: KinFu, ElasticFusion and our proposed method using “Plant” dataset where the camera motion is faster than the ones in the other datasets. The plant and the base under the plant are erroneously reconstructed for multiple times as shown in Fig. 5(a) KinFu and Fig. 5(b) ElasticFusion. Also, the desks in front of the plant become distorted in both methods due to fast camera motion. Our proposed method overcomes this shortcoming and produces the best reconstruction quality as shown in Fig. 5(c).

C. Reconstruction of Moving Objects

Fig. 6(d) shows the result of our proposed method which combines volume reconstruction for the static background with point-based reconstruction for the moving object. As shown in Fig. 6(c), the original KinFu cannot track and reconstruct the moving tissue box shown in Fig. 6(a) and its corresponding area has been reconstructed incorrectly due to the motion. However, in our proposed method, the moving object is estimated in pose estimation, reconstructed using a point-based method and then added into the current 3D static background.

Fig.7 shows the performance of region growing using

TABLE I. ATE AND RPE FOR TUM BENCHMARK BASED ON KINFU, POINT-BASED FUSION, ELASTICFUSION AND PROPOSED PIPELINE.

DATASET Name	**Number of frames**	**Absolute Trajectory Error (Unit: centimeter)**					**Relative Pose Error (Unit: centimeter)**				
		KinFu [2]	***Point-based Fusion [11]***	***Improved point averaging (Proposed)***	***Elastic Fusion [10]***	***Improved point averaging with Joint ICP (Proposed)***	***KinFu [2]***	***Point-based Fusion [11]***	***Improved point averaging (Proposed)***	***Elastic Fusion [10]***	***Improved point averaging with Joint ICP (Proposed)***
XYZ-1	790	2.232	2.364	2.296	1.562	1.577	2.681	2.803	2.739	2.434	2.436
XYZ-2	1000	2.137	1.993	1.804	0.753	0.901	2.985	2.856	2.375	1.203	1.300
DESK	1000	3.820	4.041	4.082	3.983	3.197	7.672	7.890	7.937	7.408	6.801
TEDDY	400	8.284	13.546	12.699	2.752	2.295	13.178	16.727	16.332	3.334	3.117
PLANT	250	19.891	21.753	18.153	2.638	2.127	26.702	28.216	21.269	3.459	3.198
FLOWER	1000	23.157	11.748	11.188	45.12	45.57	31.455	17.063	17.217	69.142	68.523
Improved Percentage			-4.77%	2.16%	25.05%	28.15%		1.68%	9.42%	19.04%	27.11%

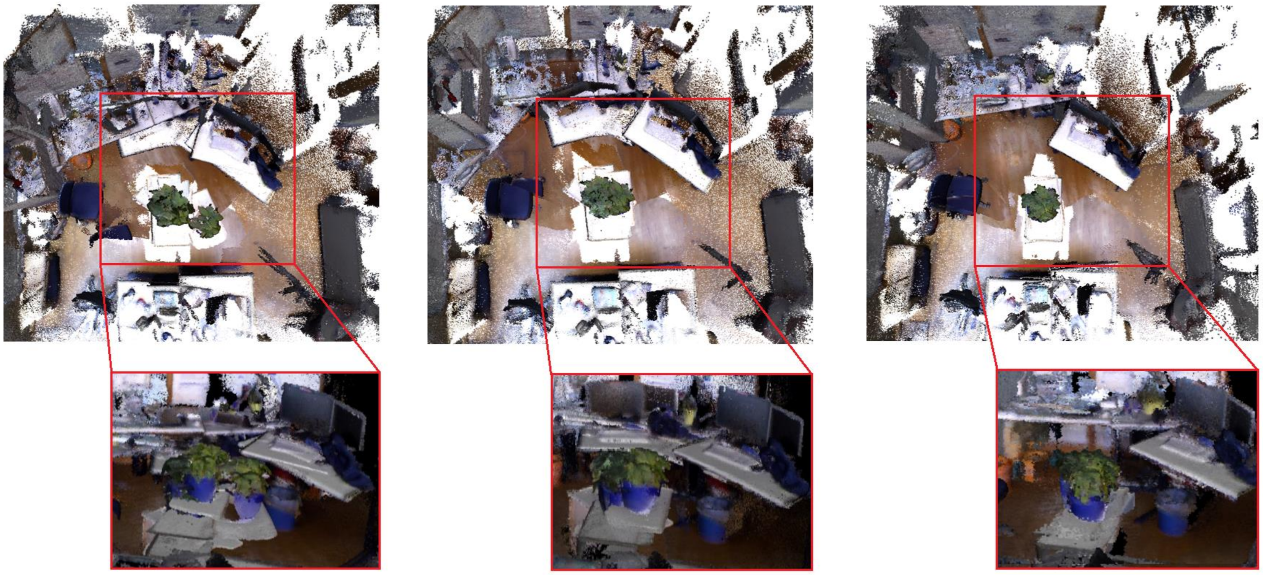
(a) (b) (c)

Figure 5. Comparison of 3D reconstruction results for “Plant” Dataset using (a) KinFu, (b) ElasticFusion, and (c) proposed method.

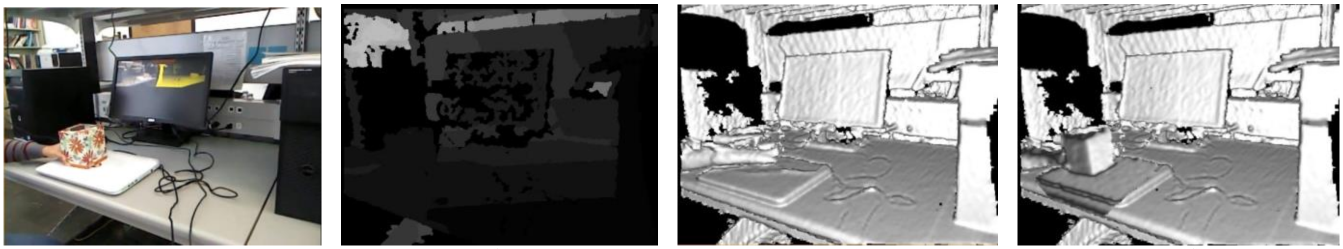
(a) (b) (c) (d)

Figure 6. Comparison between the original KinFu and the proposed method for ‘Moving_box’ dataset. (a) RGB image, (b) depth map, (c) 3D reconstruction using KinFu, and (d) 3D reconstruction using the proposed method

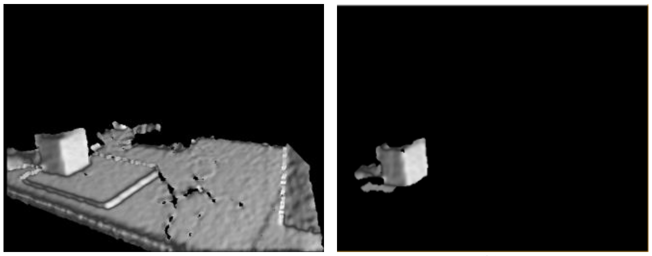
(a) (b)

Figure 7. Region growing results obtained by (a) using depth similarity only and (b) using both depth, orientation, and intensity similarity together.

different similarity criteria for segmentation. In Fig. 7 (a), the region map consists of a part of the static background getting merged with the moving object which also gets reconstructed in the 3D reconstruction of moving objects. It is because only the depth information is used as the similarity measure for segmentation. To deal with this drawback, we introduce normal map and intensity map along with the depth values as similarity measure to avoid growing into regions that do not belong to the moving object. The result for the proposed region growing method using the combined similarity criteria, which was mentioned in Section II.D, is shown in Fig. 7 (b).

TABLE II. PRECISION FOR SBM-RGBD DATASET USING PROPOSED METHOD

DATASET NAME	Number of frames	Precision	
		Point-based Fusion [11]	*Improved point averaging with Joint ICP (Proposed)*
Shelves	550	74.39%	80.76%
Abaondoned2	250	87.15%	95.02%
genSeq1	410	83.28%	86.16%

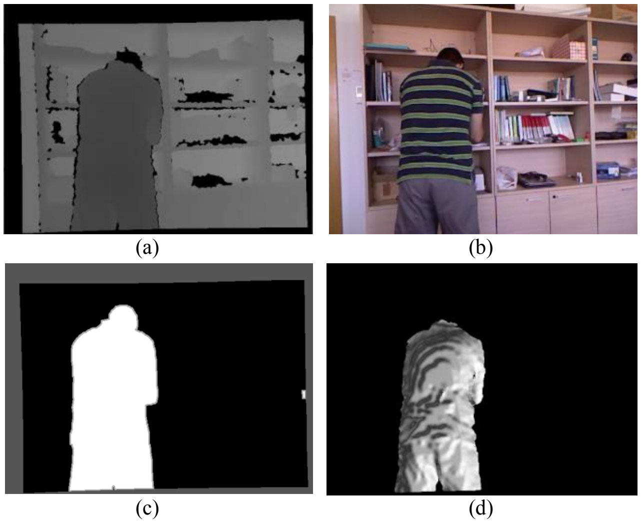
(a) (b)
(c) (d)

Figure 8. An example of incomplete 3D reconstruction due to noisy depth map: (a) depth map, (b) RGB image, (c) ground truth, and (d) 3D reconstruction of the foreground moving object.

We can observe that the moving objects only include moving tissue box and hand.

To evaluate the accurate performance of segmentation on foreground moving objects, we adopt SBM-RGBD Dataset which contains various challenge scenarios such as illumination changes, color camouflage, depth camouflage, etc. The precision metric used is given as Eq. (10):

$$precision = \frac{TP}{TP+FP}, \qquad (10)$$

where TP represents the number of the dynamic pixels correctly segmented as foreground moving objects. FP stands for the number of background pixels wrongly classified as foreground. Table II gives the results using Dataset "Shelves" where the moving object is the person. The performance using our proposed method achieves 80.76% precision and, it is better than the performance using only point-based fusion by 6.37%. According to our observations, the performance of 3D segmentation is directly affected by the input depth. In Fig. 8(a), the depth information of the head is missing because the color is close to black which causes the infrared ray emitted by the sensor to be nearly absorbed. Table II also represents the results of other experiments using SBM-RGBD Dataset.

IV. CONCLUSIONS

In this paper, we propose an improved method for real-time 3D reconstruction. It enhances sensor tracking ability and can deal with dynamic scenes by combining volumetric and point-based representation. To improve camera tracking ability and achieve better selection of moving objects' points, we define a 3D radius map which takes into account the characteristics of each depth sensor more accurately. Also, we propose a weighted ICP algorithm combined with RGB-D information to improve the stability of camera tracking. A GPU-based region growing method which considers orientation similarity and intensity level along with depth similarity has been proposed to segment moving objects more accurately. To ensure efficient use of GPU memory, detected motion objects are reconstructed by using simple point-based fusion. They are combined with voxel-based reconstruction of the static background. Both parts of the scene are reconstructed simultaneously and added into the global model without any time delay. Experimental results show that our proposed method has more stable tracking ability compared to some other state-of-the-art methods. Currently, the foreground segmented objects in our proposed method do not share coherent information. It fails to track the moving objects in the temporal domain. Our future work will focus on this challenge to realize tracking dynamic objects.

REFERENCES

[1] 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 *Proc. ACM Symposium User Interface Software and Technology (UIST)*, pp. 559-568, Oct. 2011.

[2] Open Perception Foundation. (2015). Point Cloud Library Homepage. Available: http://point clouds.org/.

[3] J. Smisek, M. Jancosek, and T. Pajdla, "3D with Kinect," in *Proc. IEEE International Conference on Computer Vision Workshops*, pp. 1154-1160, Nov. 2011.

[4] C. Tomasi and R. Manduchi. "Bilateral Filtering for Gray and Color Images," in *Proc. International Conference on Computer Vision (ICCV),* 1998.

[5] P. Besl and N. McKay. "A Method of Registration of 3D Shapes," in *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI),* pp. 239-256, 1992.

[6] B. Curless and M. Levoy. "A Volumetric Method for Building Complex Models from Range Images," in *Proc. Computer Graphic and Interactive Techniques*, pp. 303-312, 1996.

[7] S. Parker, P. Shirley, Y. Livant, C. Hansen and P. Sloan. "Interactive Ray Tracing for Isosurface Rendering," in *Proc. of Visualization,* 1998.

[8] K. Khoshelham. "Accuracy Analysis of Kinect Depth Data," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVIII-5/W12, 2011.

[9] R. R. Srinivasan, Z. Xia, J. Kim and Y. S. Park, "Confidence Indicators Based Pose Estimation for High-quality 3D Reconstruction Using Depth Image," in *Proc. Visual Communications and Image Processing (VCIP),* Dec. 2015.

[10] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker and A. J. Davison, "ElasticFusion: Dense SLAM Without A Pose Graph**,"** in *Robotics: Science and System (RSS)*, 2015.

[11] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich and A. Kolb. "Real-time 3D Reconstruction in Dynamic Scenes using Point-based Fusion," in *Proc. International Conference on 3D Vision (3DV)*, pp. 1-8, 2013.

[12] T. Weise, T. Wismer, B. Leibe and L. Van Gool. "In-hand Scanning with Online Loop Closure," *in Proc. IEEE International Conference Computer Vision Workshops*, pp. 1630-1637, 2009.

[13] M. Zwicker, H. Pfister., J. V. Baar, and M. Gross. "Surface Splatting," in *Procs. SIGGRAPH*, pp. 371-378, 2001.

[14] J. Jung, J.Y. Lee, Y. Jeong and I. S. Kweon. "Time-of-Flight Sensor Calibration for a Color and Depth Camera Pair," in *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 37, No. 7, July, 2015.

[15] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *Proc. International Conference Intelligent Robot System*, pp. 573–580, 2012.

[16] M. Camplani, L. Maddalena, G. Moyà Alcover, A. Petrosino and L. Salgado, "A Benchmarking Framework for Background Subtraction in RGBD videos," in *Image Analysis and Processing (ICIAP)*, Springer, 2017