Proceedings of the 2017 IEEE
International Conference on Robotics and Biomimetics
December 5-8, 2017, Macau SAR, China

Invisibility: A Moving-object Removal Approach for Dynamic Scene Modelling using RGB-D Camera

Yuxiang Sun[1], *Student Member, IEEE,* Ming Liu[2], *Member, IEEE,* and Max Q.-H. Meng[1], *Fellow, IEEE*

***Abstract*— Scene modelling is of great importance for robots in unknown environments. Existing Visual Simultaneous Localization and Mapping (Visual SLAM) approaches are able to build impressive scene models using RGB-D cameras in static scenes. In dynamic scenes, however, moving objects can be recorded as spurious objects, which contaminates the resulting scene models. In order to build clear scene models, we propose a novel moving-object removal approach for scene modelling algorithms in this paper. Our approach does not rely on prior knowledge, such as appearance features or initial segmentation. In addition, the proposed approach does not require an initialization process, which is different from most background subtraction algorithms. The experimental results demonstrate that our approach is able to effectively remove moving objects and assist scene modelling algorithms to build clear models in dynamic scenes.**

I. Introduction

Affordable-cost RGB-D cameras have raised great interests for researchers since their development [1]. Compared with traditional 3-D range finders, such as laser-scanners, RGB-D cameras provide rich visual information. Many effective scene modelling solutions, for instance, Visual Simutaneous Localization and Mapping (Visual SLAM) [2]–[7] have been proposed. They can build impressive scene models in static scenes. In dynamic scenes, however, moving objects can be recorded as spurious objects, which contaminates the resulting scene models. Fig. 1 shows a point-cloud scene model built by an Iterative Closet Point (ICP)-based modelling algorithm in a dynamic scene. We can see that the persons are recorded as spurious objects in the resulting model. The contaminated scene model becomes useless for future applications. Matching the spurious areas will give ambiguous decisions [8]. For instance, if a robot returns to a location where some moving objects are recorded in the model while the current scan does not contain any feature corresponding to the moving objects, the scene recognition will be confused.

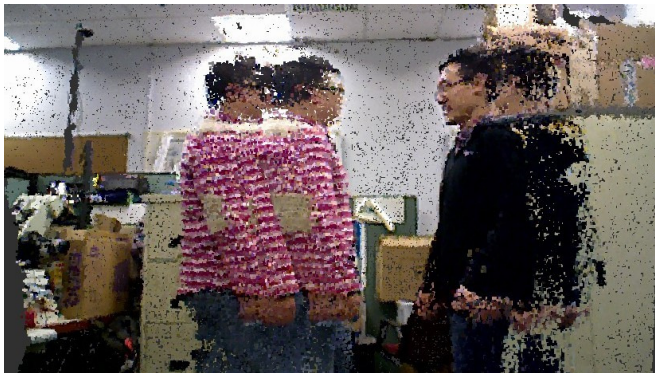

Fig. 1: The contaminated scene model built by ICP fusion. In this scene, two persons are walking in opposite directions in front of a static RGB-D camera.

[1]Yuxiang Sun and [1]Max Q.-H. Meng are with the Shenzhen Research Institute, The Chinese University of Hong Kong, Shenzhen, Guangdong, China; Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China. email:{yxsun,qhmeng}@ee.cuhk.edu.hk

[2]Ming Liu is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China. email:eelium@ust.hk

Moving objects are unavoidable interferences in real world scenes. In order to build clear scene models, we propose a novel approach to remove moving objects from RGB-D frames. The proposed approach consists of two stages: the motion detection stage and the motion segmentation stage. The RGB-D frames with moving objects filtered are taken as inputs for ICP fusion. We employ walking persons as moving objects in the experiments. The results demonstrate that our approach has improved performance of moving-object removal compared with state-of-the-art methods. The point-cloud models built with our approach contain moving-object points at the minimum level.

The work with similar motivation to ours is the method proposed by Litomisky *et al* [9]. They firstly found corresponding objects in two views of a scene. Then, they identified moving objects by checking whether the position of the corresponding objects changes between the two views. If the position changed, they concluded that the object was moving and should be removed in the point-cloud frames.

The remainder of this paper is organized as follows. In section II, we present the overview of our approach. In section III, details of our approach are discussed. In section IV, experimental results are analysed. In the last section, we conclude this paper and discuss the future work.

II. Overview of Approach

This section overviews the proposed moving-object removal approach. We use the Asus Xtion Pro Live to record data. The color and the depth data provided by the RGB-D camera are both used. The color data are used in the detection stage. The depth data are used in the segmentation stage. The results from the motion detection stage provide likelihood information for the motion segmentation stage.

Fig. 2 illustrates the schematic overview of our approach. Firstly, we project the organized RGB-D point-cloud frames into 2-D RGB and depth images. Secondly, we apply the

978-1-5386-3742-5/17/$31.00 © 2017 IEEE

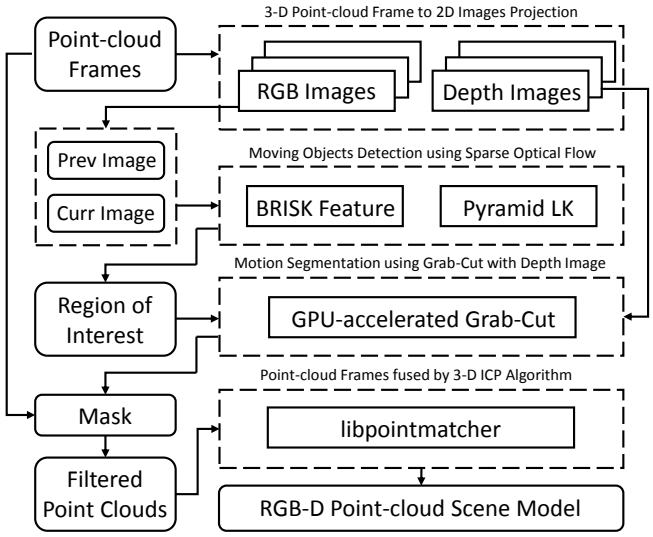

Fig. 2: The schematic overview of our approach. The `Prev Image` is the previous RGB image captured at time $t-1$. The `Curr Image` is the current RGB image captured at time t.

phyramid Lucas-Kanade optical flow algorithm [10] with BRISK feature [11] against two consecutive RGB images. Moving objects are roughly detected by thresholding the optical flow values. We create the rectangular contour from the moving-object feature points. Thirdly, the rectangular contour is used as the Region-of-Interest (ROI) which provides the motion likelihood for the Grab-Cut based motion segmentation [12] algorithm to segment the moving objects. Finally, the point-cloud frames with moving objects filtered are taken as input for ICP fusion to build the scene model. We use the open-source ICP algorithm library `libpointmatcher` [13] in this paper. It should be noted that the RGB-D camera does not move during the scene modelling process in this paper.

III. Moving-object Removal

A. Moving-object Detection

Optical flow is a kind of motion detection algorithm [14] that has been extensively studied over the past decades. The general idea of optical flow is to determine the point correspondences from two consecutive images under the assumption of spatial-temporal consistency of the images. The optical flow problem can be solved by dense and sparse solutions. The dense solution computes optical flow over all pixels in the image and find flow values for each pixel. The sparse one just computes the flow vectors on some points of interests. The optical flow problem can be formulated as follows:

$$\mathcal{J}(d_x,d_y)=\sum_{(x,y)\in\mathcal{N}}\left[I(x,y,t-1)-I(x+d_x,y+d_y,t)\right]^2, \quad (1)$$

where $\mathcal{N}$ is the set of neighbour pixels of point (x_0,y_0) which is the center of the patch of pixels, $I(x,y,t-1)$ is the pixel intensity value of (x,y) in the $t-1$ image frame, $I(x+d_x,y+d_y,t)$ is the corresponding pixel intensity value in the t image frame. This equation proposes a cost function $\mathcal{J}(d_x,d_y)$ for a quadratic optimization problem. The flow vector (d_x,d_y) of the point (x_0,y_0) can be found by minimizing $\mathcal{J}(d_x,d_y)$ over the patch of pixels.

(a) (b)

Fig. 3: The demonstration for the motion detection. In this scene, a person is walking with a normal speed in front of a static RGB-D camera. (a) and (b) show the motion detection results at different time. The feature points from the walking person and the background are coloured as green and red. The areas inside the white rectangles are the created ROIs.

In this paper, we adopt the widely used sparse optical flow method proposed by Lucas-Kanade [15] to determine the flow values for extracted feature points. However, the standard Lucas-Kanade method requires small pixel displacements. Thus, we use the pyramidal implementation to determine the flow vectors at different scales. The moving objects can be roughly determined depending on a predefined tolerance τ by the following inequalities:

$$\begin{aligned} d &> \tau, \quad f_i \in \mathcal{F}_{moving} \\ d &< \tau, \quad f_i \in \mathcal{F}_{static} \end{aligned} \quad (2)$$

where $d=\sqrt{d_x^2+d_y^2}$ is the $2-norm$ of the flow vector of the feature point f_i, $\mathcal{F}_{moving}$ and $\mathcal{F}_{static}$ are moving and static feature point sets. The points in $\mathcal{F}_{moving}$ are tracked in the consecutive frames. We eliminate the point set $\mathcal{F}_{static}$ after the optical flow computation. The feature points are re-extracted randomly to ensure the pre-defined amount of feature points.

Fig. 3 shows the results of motion detection in a dynamic scene. We can see that the feature points from the moving person are tracked at different frames. Note that we set the width of the ROI to the width of the rectangular contour of the tracked moving-object points. The height of the ROI is set to the height of the image. The areas inside the ROIs are the possible foregrounds. The areas outside the ROIs are considered as the static backgrounds. The ROIs provide the motion likelihood for the motion segmentation.

B. Moving-object Segmentation

In this section, we pixel-wisely segment the foreground using Grab-Cut [12] with the motion likelihood provided by the moving-object detection. Grab-Cut is an enhanced version of Graph-Cut [16]. It employs an iterative optimization method and requires no complete labelling for user inputs. Similarly

as the Graph-Cut, it models the random variables with user inputs and image data in a Conditional Random Field (CRF), and uses the Min-Cut algorithm [17] to solve the energy minimization problem. CRF is a kind of Markov Random Field (MRF). The only difference between CRF and MRF is that the CRF also depends on the image data [18], such as pixel intensity values. The flowing equation illustrates the conditional independency and markovianity of CRF:

$$p(l_{x_i}|I, l_{X-\{x_i\}}) = p(l_{x_i}|I, l_{\mathcal{N}_{x_i}}), \tag{3}$$

where the random variable is the label l_{x_i} for pixel x_i, I is the intensity set of all pixels X in the depth image, $l_{X-\{x_i\}}$ represent the labels except l_{x_i}, $\mathcal{N}_{x_i}$ is the neighbour set of pixel x_i. We adopt the classical 4-neighbour configuration in this paper. The label distribution can be represented in the following equation:

$$p(l_{x_i}|I) = \frac{1}{Z}\exp\big(-\frac{1}{T}E(l_{x_i}|I)\big), \tag{4}$$

where T is a constant, $Z = \sum_{x_i \in X}\big(-\frac{1}{T}E(l_{x_i}|I)\big)$ is the partition function. The energy function for all labels $\mathcal{E}(L|I)$ consists of unary potentials and pairwise potentials:

$$\mathcal{E}(L|I) = \sum_{x_i \in X}\psi_i(l_{x_i}) + \sum_{x_i \in X, x_j \in \mathcal{N}_{x_i}}\psi_{ij}(l_{x_i}, l_{x_j}|I), \tag{5}$$

where $l_{x_i} \in L$, $\psi_i(l_{x_i})$ is the unary potential which is determined by the likelihood information, $\psi_{ij}(l_{x_i}, l_{x_j}|I)$ is the pairwise potential which depends on the pixel intensity values. The pairwise potential is also called the smoothness term in CRF. It can be modelled using the following equation:

$$\psi_{ij}(l_{x_i}, l_{x_j}|I) \propto \alpha\exp\big(-\beta\|I(x_i) - I(x_j)\|^2\big), \tag{6}$$

where $I(x_i)$ and $I(x_j)$ are pixel intensity values of x_i and x_j, α and β are real number parameters.

The major advantage of Grab-Cut is that the unary potentials are derived from Gaussian Mixture Models (GMM). The unary potentials can be dynamically updated with GMM in each iteration. Suppose that we create $\mathcal{K}$ clusters in the depth image, the total number of GMM components is $2\mathcal{K}$. The probability for pixel x_i being the foreground can be expressed as follows:

$$p(x_i \in \mathcal{FG}) = \sum_{\kappa=1}^{\mathcal{K}} \pi_\kappa \mathcal{N}(x_i|\mu_\kappa, \Sigma_\kappa). \tag{7}$$

where μ_κ and Σ_κ are determined by the pixel values of the corresponding clusters. Similarly, the probability of pixel x_i being the background $p(x_i \in \mathcal{BG})$ can be expressed with another set of GMMs. The unary potentials of x_i are modelled using the negative logarithm of the probabilities:

$$\begin{aligned}\psi_i(l_{x_i} = 255) &= -\log p(x_i \in \mathcal{FG}) \\ \psi_i(l_{x_i} = 0) &= -\log p(x_i \in \mathcal{BG})\end{aligned}, \tag{8}$$

where $l_{x_i} = 255$ and $l_{x_i} = 0$ represent the labels for foreground and background pixels in the mask image respectively.

Fig. 4 compares the segmentation results using RGB and depth images with a same ROI. We can see from this qualitative demonstration that the segmentation result can be greatly improved by using the depth image. We think the reason is that the pixel values of the walking person in depth images are more consistent, which is a benefit for the image segmentation algorithm.

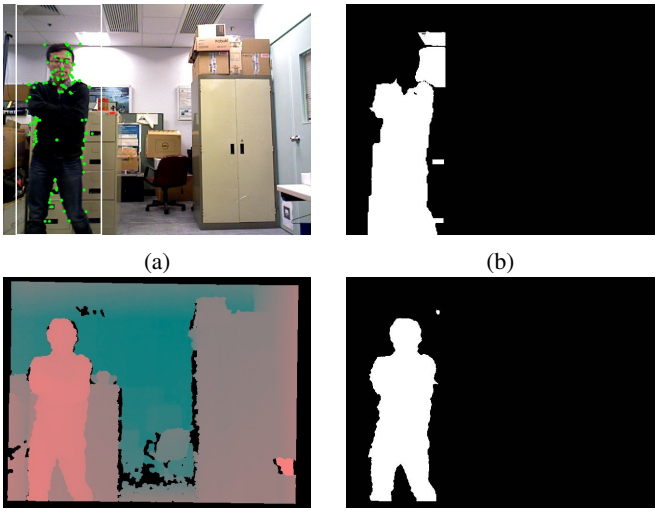

Fig. 4: The comparison of the segmentation results using the RGB and depth images. (a) shows the moving-object detection result and the created ROI. (b) is the segmentation result using the RGB image with the ROI. (c) is the depth image (distance increasing from red to green). (d) is the segmentation result using the depth image with the same ROI. A visible improvement is demonstrated by the comparison.

IV. Experimental Results and Discussions

A. Experimental Setup

We performed the experiments in three different scenes. Scene I is a normal office room. Scene II is also an office room, but there exhibit some glass and monitor surfaces which can cause unstable depth measurements due to the shiny surfaces. Scene III is a common indoor corridor. Two walking persons were employed as moving objects in the experiments.

Our program runs on a PC with an Intel i3 CPU, 4GB memories and an NVIDIA GTX770 GPU. The RGB-D images are captured at around 30Hz at the resolution of 640×480. The average time cost for the GPU accelerated Grab-Cut algorithm on one frame is about 100ms without code optimization.

B. Moving-object Removal Demonstrations

Fig. 5 qualitatively demonstrates the moving-object removal results. We can see that the walking persons are successfully removed and filtered from the corresponding RGB-D point-cloud frames. However, we find that the contours of the walking persons in the segmentation results are not smooth. We think this is caused by the unstable depth measurements at object boundaries.

We qualitatively compared the proposed moving-object removal approach with background subtraction algorithms:

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

Fig. 5: The moving-object removal demonstration. From the top row to the bottom row, the experiments are performed in scenes I, II and III respectively. Column (a) are the RGB images with the moving-object detection results. Column (b) are the corresponding depth images. Column (c) are the moving-object segmentation results. Column (d) are the RGB-D point-cloud frames with the walking persons filtered. Note that the depth-unavailable areas in column (d) are denoted with the same color as the moving objects.

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

Fig. 6: The qualitative comparison for the moving-object removal. The shown scenario is a person walking in an office room. (a) shows the RGB image. (b)-(e) are the moving-object removal results provided by our approach, GMM, ViBe and PBAS respectively. The improved performance of our approach is revealed by the comparison.

GMM [19], ViBe [20] and PBAS [21]. GMM is a kind of parametric method. ViBe and PBAS are the state-of-the-art non-parametric methods. The background subtraction algorithms take as input the RGB images. The comparison results are presented in Fig. 6. We can find that our method greatly outperforms the others. From Fig. 6 (c)-(e), we can see there are lots of incorrectly labelled pixels. These labels can lead to failures of moving-object removal from point-cloud frames.

C. Scene Modelling Demonstrations

The RGB-D point-cloud frames with moving objects filtered are taken as input for ICP fusion to build the scene models. We compare the modelling process of our approach with those of the background subtraction algorithms. The RGB-D point-cloud frames with moving objects filtered by these approaches are individually fed into the `libpointmatcher`.

The processes of the scene modelling are demonstrated in Fig. 7. From the sub-figures of the top row, we can see that the unavailable areas caused by moving-object removal are gradually complemented from adjacent frames by the ICP algorithm. We also find that very few points from the walking person are recorded in the model. This demonstrates that our approach is able to assist ICP algorithm to build clear point-cloud models in dynamic scenes. The sub-figures of

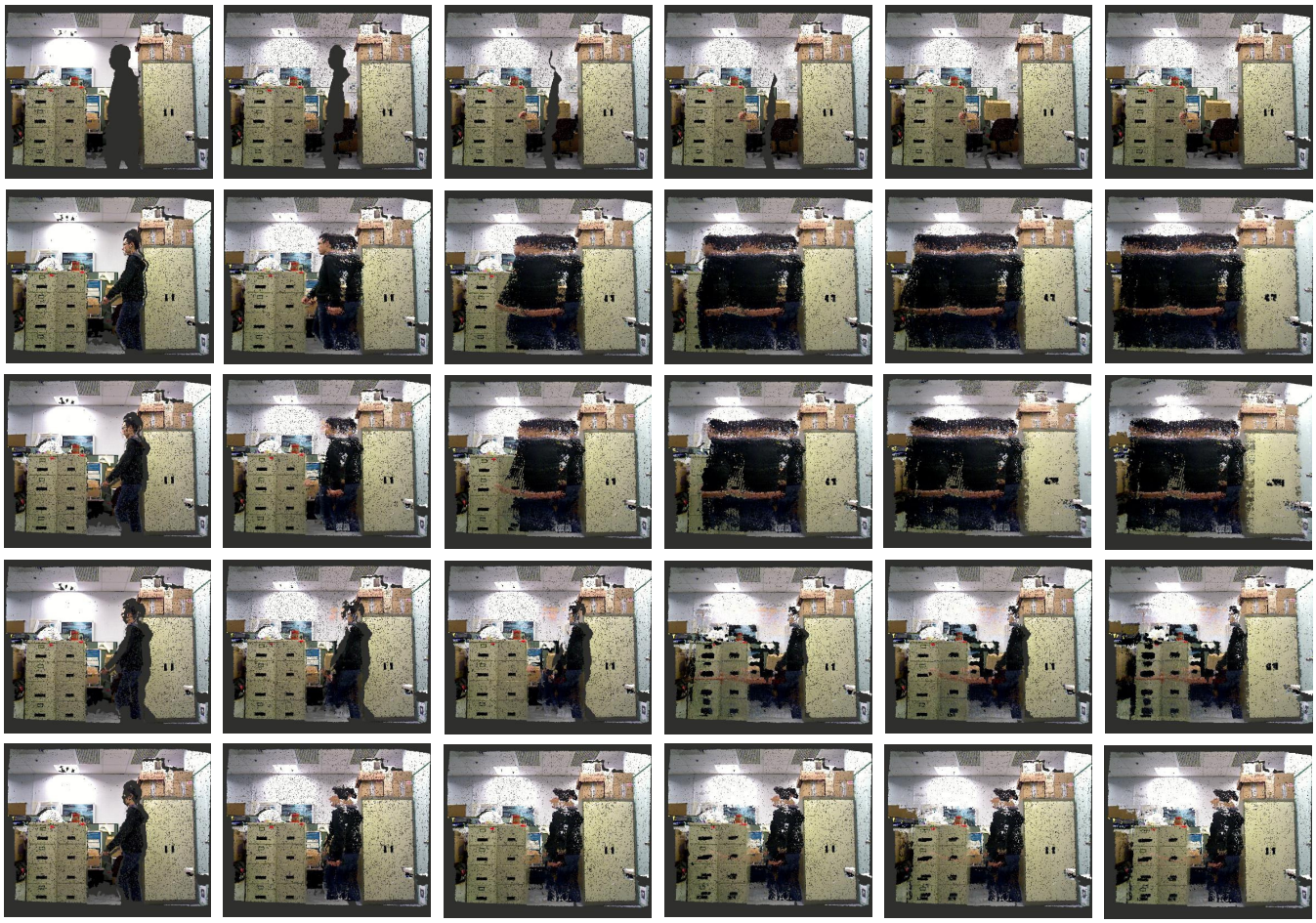

Fig. 7: The qualitative comparison for the scene modelling processes. The shown scenario is a person walking in an office room. The sub-figures from the top row to the bottom row are samples snapshots from the scene modelling processes of our approach, the original `libpointmatcher` without moving-object removal, GMM, ViBe and PBAS respectively. The processes develop chronological from left to right. The last column of sub-figures present the final resulting point-cloud models. The enhanced performance of our proposed method is demonstrated by the comparison.

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

Fig. 8: Bird's-eye view of the final resulting point-cloud models. The remaining moving-object points that are recorded in the models are highlighted with red ellipses. (a)-(e) are the point-cloud models built by our approach, the original `libpointmatcher`, GMM, ViBe and PBAS respectively. The optimal performance of our proposed approach is revealed by the comparison.

the second row show the ICP fusion without moving-object removal. We can see that the walking person is gradually recorded as spurious objects in the point-cloud model. The recorded spurious points contaminate the resulting scene model. The sub-figures of the third row show the results of the GMM algorithm. Because there are so many false negatives in the segmentation results, the built model is almost the same as the results given by the ICP fusion without moving-object removal. The sub-figures of the 4th and the bottom rows are the results of the ViBe and PBAS algorithms. Because they have similar segmentation performance, we can see that the built point-cloud models are very similar. The

models built by the background subtraction algorithms are practically useless because the moving-object points largely jeopardize the visibility of the scene representation.

Fig. 8 presents the bird's-eye view for the final resulting point-cloud models. We can clearly find that our approach is able to provide a clear scene model with the remaining moving-object points at the minimum level. The comparison demonstrates that our approach greatly outperforms the state-of-the-art methods.

V. Conclusions

In this paper, we proposed a novel approach for moving-object removal in dynamic scenes. The experimental results demonstrate that our approach is able to effectively remove moving objects and assist scene modelling algorithms to build clear models that contains moving-object points at the minimum level. However, our approach assumes that the camera is static during the scene modelling process. This seemingly limits our approach to be used efficiently on moving platforms. However, by encoding the pose estimations using the extracted static points, the proposed approach is still valid accordingly to our recent test. The result will be reported in the future.

Acknowledgment

Research presented in this paper was supported by the Shenzhen Science and Technology Program Project No. JCYJ20170413161616163 awarded to Prof. Max Q.-H. Meng. The authors would like to thank Zhe Min and Lin Qi for the preparation of the experiments.

References

[1] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE transactions on cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[2] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136.

[3] 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," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

[4] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[6] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[7] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, 2017.

[8] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robotics and Autonomous Systems*, vol. 89, pp. 110 – 122, 2017.

[9] K. Litomisky and B. Bhanu, "Removing moving objects from point cloud scenes," in *Advances in Depth Image Analysis and Applications*. Springer, 2013, pp. 50–58.

[10] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, no. 1-10, p. 4, 2001.

[11] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2548–2555.

[12] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[13] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[14] A. Agarwal, S. Gupta, and D. K. Singh, "Review of optical flow technique for moving object detection," in *Contemporary Computing and Informatics (IC3I), 2016 2nd International Conference on*. IEEE, 2016, pp. 409–413.

[15] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International journal of computer vision*, vol. 56, no. 3, pp. 221–255, 2004.

[16] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," in *ACM Transactions on Graphics (ToG)*, vol. 22, no. 3. ACM, 2003, pp. 277–286.

[17] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[18] S. Kumar and M. Hebert, "Discriminative random fields," *International Journal of Computer Vision*, vol. 68, no. 2, pp. 179–201, 2006.

[19] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2. IEEE, 2004, pp. 28–31.

[20] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image processing*, vol. 20, no. 6, pp. 1709–1724, 2011.

[21] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 38–43.