Proceedings of the 2015
IEEE Conference on Robotics and Biomimetics
Zhuhai, China, December 6-9, 2015

Motion Removal from Moving Platforms: An RGB-D Data-based Motion Detection, Tracking and Segmentation Approach

Yuxiang Sun[1], Ming Liu[2] and Max Q.-H. Meng[3], *Fellow*, *IEEE*

***Abstract*— Moving objects are the primary concern for most robot vision applications in dynamic environments. The existence of moving objects can lead to ambiguous decisions such as searching loop closure in visual mapping applications. Eliminating moving objects from the image sequences captured by moving camera is the key challenge. In this paper, a novel approach for moving objects removal using a hand-held RGB-D camera is proposed. Only the visual and depth data are used. No other sensor or prior information is needed in this paper. Our approach can be exploited as a pre-processing stage to filter out data that are associated with moving objects. We test our approach with various ego-motion patterns in different environments. The experimental results demonstrate that our approach can provide a practical solution for motion removal from moving platforms using an RGB-D camera.**

I. Introduction

For robot vision applications, such as visual odometry, navigation and mapping[1]-[2], plenty of effective approaches and algorithms have been proposed. However, most of the existing solutions were developed under the assumption of static environments. In dynamic environments, the existence of moving objects can lead to unstable results or even failures. For instance, comparing the scene with and without moving objects can corrupt the loop closing in visual mapping. To reliably work in dynamic environments requires robots being able to separate moving objects from static backgrounds. Thus, it is of great significance for motion removal from the image sequences captured by cameras on moving platforms.

The motions in image sequences captured by a moving camera are caused by camera ego-motion and moving objects. In order to eliminate the moving objects, the key information is the ego-motion of the camera. With the camera ego-motion, the moving objects can be detected by motion compensation approaches. The ego-motion estimation merely using visual sensors is usually called visual odometry. It is an active topic which has attracted lots of researchers in robot vision. The general idea of visual odometry is to derive camera poses and orientations from associated images. Most early approaches were developed based on feature tracking techniques. The transformations can be found by minimizing the cost function of the re-projection error between the feature pairs in consecutive frames. Recent approaches are based on pixel intensity values. These approaches require the assumption of brightness consistency of pixels in consecutive frames. The ego-motion can be found by minimizing the photometric errors.

With the camera ego-motion, it is possible to detect moving objects using the ego-motion compensation technique. Kim *et al.* proposed a non-panoramic background modelling approach[3] to detect moving objects from moving platforms. The background can be modelled using pixel-wise spatio-temporal Gaussian models from warped images. The moving objects can be detected in the overlapped areas using the background subtraction method.

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

II. Overview of Approach

In this paper, we proposed an RGB-D data-based motion detection, tracking and segmentation approach. Our approach is an online framework. A particle filter-based tacking mechanism plays a central role in this paper. In our method, we purely use the visual and depth data from the RGB-D camera. No other sensors such as IMU or prior information is required. We adopt the similar motion detection and tracking ideas proposed by Jung *et al.*[4] in this paper.

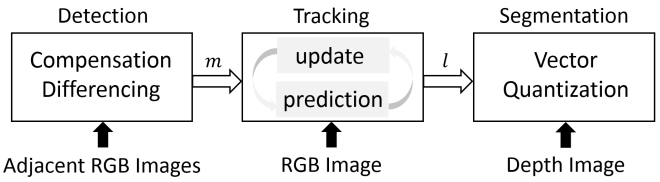

Fig. 1. The overview of our approach. The ego-motion compensation and the frame differencing are the key operations in the *detection* stage. m denotes the measurement information for the particle filter from the *detection* stage, l denotes the likelihood information for the MAP estimation from the *tracking* stage.

As shown in Fig.1, we use the RGB images for moving objects detection and tracking. The camera ego-motion[5] is derived from two consecutive RGB images using the RANSAC algorithm[6]. Note that the camera ego-motion is estimated in 2-D image plane, which is convenient for the moving objects detection in our approach. The moving objects are detected by subtracting the ego-motion compensated frame at time $t-1$ with the frame at time t. The pixel values of the differencing image provide the measurement information for

[1]Yuxiang Sun is a PhD student and [3]Max Q.-H. Meng is a Professor. They are both at the Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T. Hong Kong SAR, China. email:{yxsun,qhmeng}@ee.cuhk.edu.hk

[2]Ming Liu is a Assistant Professor at the Department of Mechanical and Biomedical Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong SAR, China. email:mingliu@cityu.edu.hk

978-1-4673-9675-2/15/$31.00 © 2015 IEEE

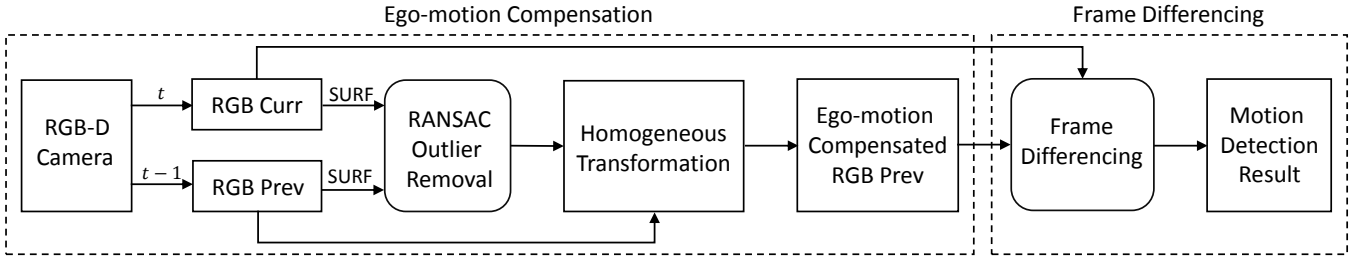

Fig. 2. The schematic diagram of the ego-motion compensation and the frame differencing techniques. The *RGB Prev* and *RGB Curr* represent the previous and current images that are captured at time $t-1$ and t. We use the RANSAC algorithm to eliminate outliers. Moving objects are detected by subtracting the *RGB Curr* with the compensated *RGB Prev*.

the particle filter. The particle filter improves the robustness of the motion detection. The vector quantized depth image at time t is employed for the image segmentation. With the posterior belief from the *tracking* stage as the likelihood, we can use the Maximum-a-posterior (MAP) estimation method to find the cluster that has the highest foreground probability. The segmented foreground cluster is treated as the moving object in our approach.

III. Detection, Tracking and Segmentation

A. Frame Differencing

In the case of static camera, the background in the consecutive frames remains not changed. All the movements in the image sequences are caused by moving objects. When subtracting the current frame with the previous frame, the static background will be removed. The moving objects can be indicated by the remaining pixel values.

$$I_d(x,y,t) = |I(x,y,t) - I(x,y,t-1)|. \tag{1}$$

As shown in equation (1), x and y are the pixel coordinates, I_d is the intensity value of the pixel in the differencing image. The value of $I_d(x,y,t)$ is the absolute difference of a pixel value at time t and time $t-1$. The pixels that belong to moving objects can be indicated by the image subtraction results,

$$\begin{aligned} &\{(x,y) \in \{Possible\ Foreground\} \mid I_d(x,y,t) > 0\} \\ &\{(x,y) \in \{Background\} \mid I_d(x,y,t) = 0\} \end{aligned} \tag{2}$$

Equation (2) discriminates the moving objects and the static background from the pixel intensity values of the differencing image. We consider the pixel belonging to the static background if the intensity value is zero. Higher pixel value corresponds to higher foreground probability.

B. Ego-motion Compensated Frame Differencing

However, the frame differencing cannot work when the camera is not static. The idea of this paper is to compensate the previous image so that the frame differencing works like in a static platform. Because the ego-motion is estimated in 2-D image planes, we warp the image at time $t-1$ using the 2-D perspective transformation computed from the RANSAC inliers. Let $T_{t-1}^{t} \in SE(2)$ denote the 2-D perspective homogeneous matrix,

$$T_{t-1}^{t} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \in \mathbb{R}^{3\times 3}, \tag{3}$$

where a_{ij} is a real number. This matrix represents the most general form for 2-D perspective transformation.

As shown in Fig.2, we use the RANSAC algorithm[6] to eliminate outliers. The inlier pairs of SURF feature points are employed to compute the transformation caused by the camera ego-motion. The feature points are matched using the SURF descriptors. We treat the feature associations from the background as inliers, and the associations from moving objects or between moving objects and the background as outliers. The number of inliers is expected to be larger than the number of outliers. To find the homogeneous transformation matrix, we need at least 4 feature associations. Let S_a denote the minimum set of feature associations required to calculate the transformation matrix. S_A denotes the set of all the feature associations. In our case, $|S_a| = 4$ and $|S_A| = N$, where N is the total number of feature associations. Let k_{max} denote the maximum number of iterations to find the optimum transformation. k_{max} can be found by the following equation,

$$k_{max} = \frac{\log p_2}{\log(1 - p_1^{|S_a|})}, \tag{4}$$

where p_1 is the probability of a randomly selected feature association being part of a good model, p_2 is the probability of k_{max} number of consecutive failures, there is $p_2 = (1 - p_1^{|S_a|})^{k_{max}}$. We normally have $p_2 < 0.01$ and $p_1 > 0.1$, so the number of iterations $k_{max} << C_N^{|S_a|}$. Equation (4) can avoid trying every set of associations, so the computation cost can be reduced. Given a minimum set of associations, we can find the transformation matrix T_{t-1}^{t} by solving a set of linear equations. The position of a pixel (x,y) in the previous frame can be transformed to (u,v) in the next frame. The transformed coordinates are rounded to be integers. The ego-motion compensated frame is denoted as $I^T(x,y,t-1)$. The valid regions of the transformed frame are not the same as the ones of the original frame. For example, the translation will leave the regions near borders invalid. We fill these areas black. In some cases, the scene is zoomed in or zoomed

out. This may cause gaps or multiple pixel values at a same location. We apply interpolation algorithms or take average values in those cases. In each iteration, the transformation T_{t-1}^{t} is evaluated through all feature associations. The probability of an association a belonging to the inlier set S_{in} is defined as follows,

$$p(a \in S_{in}) = \exp\big(-\frac{d^2}{\sigma^2}\big), \tag{5}$$

where $d = \sqrt{(u-x)^2+(v-y)^2}$, σ is a small real number. The probability is depending on the distance between the two coordinates. We consider if $p(a \in S_{in}) > p_0$ the association $a \in S_{in}$, otherwise $a \in S_{out}$. p_0 is a pre-defined probability value. We record the number of inlier associations n in each iteration. If $n > \varepsilon$, we refine the transformation T_{t-1}^{t} using all the identified inlier associations and terminate the program, otherwise, continue the iteration and dynamically update the maximum iteration number according to n, ε is a pre-defined threshold. The refined transformation can be found by minimizing the sum of distances for all the associations in S_{in}. We define an optimization problem for this refinement,

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\sum_{i=1}^{|S_{in}|}\big[(u_i-x_i)^2+(v_i-y_i)^2\big]^2, \\ &\text{subject to} \quad (u_i,v_i) \in F_{prev}^{T},\ (x_i,y_i) \in F_{curr} \end{aligned} \tag{6}$$

where F_{prev}^{T} and F_{curr} are the transformed feature points set in the previous frame and the corresponding feature points set in the current frame, $|S_{in}|$ is the number of the inlier feature association set.

With the ego-motion compensated frame $I^T(x,y,t-1)$, the differencing image can be calculated as follows,

$$I_d(x,y,t) = |I(x,y,t) - I^T(x,y,t-1)|, \tag{7}$$

where (x,y) are the coordinates in the current frame. According to equation (2), the possible foreground can be identified according to the pixel values of the differencing image. Fig.3 demonstrates the sample results of the ego-motion compensated frame differencing method. In this scenario, a person is walking straightly in an office room. The positions of background pixels in (c) are similar as those in (b). The blank areas caused by the transformation are filled black. The moving objects can be indicated by the subtraction results.

C. Particle Filter-based Tracking

Let $\boldsymbol{x}$ denote the state variable of the particle filter. It includes the particle positions and velocities. The state variable of a particle i at time t is represented as follows,

$$\boldsymbol{x}_{i,t} = [x_{i,t},\ y_{i,t},\ \dot{x}_{i,t},\ \dot{y}_{i,t}]^T, \tag{8}$$

where x and y are the coordinates of the particle, $\dot{x}$ and $\dot{y}$ are the velocities along the two coordinate axes. The state variable at time t can be predicted by the velocities and the prior belief at time $t-1$. With the measurement information z_t and the prior belief $\overline{bel}(\boldsymbol{x}_{i,t})$ from the prediction stage, the posterior belief $bel(\boldsymbol{x}_{i,t})$ can be estimated by the Bayesian filter[7]. Let $p(\boldsymbol{x}_{i,t}|\boldsymbol{u}_{i,t},\boldsymbol{x}_{i,t-1})$ denote the transition probability from time $t-1$ to t, $p(z_{i,t}|\boldsymbol{x}_{i,t})$ denote the measurement probability at time t. We have the following Bayesian recursive equations,

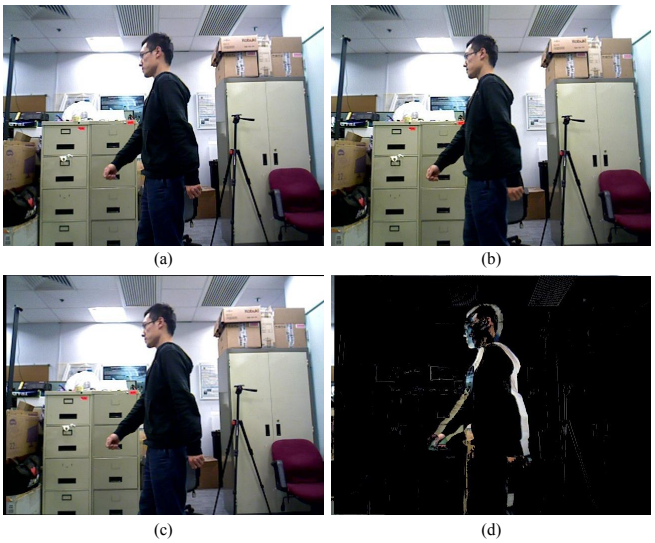

Fig. 3. The sample results of the ego-motion compensated frame differencing method. (a) and (b) are the images captured at time $t-1$ and t. (c) is the warped version of (a). (d) is differencing image between (b) and (c).

$$\begin{cases} \overline{bel}(\boldsymbol{x}_{i,t}) = \sum p(\boldsymbol{x}_{i,t}|\boldsymbol{u}_{i,t},\boldsymbol{x}_{i,t-1})bel(\boldsymbol{x}_{i,t-1}) \\ bel(\boldsymbol{x}_{i,t}) = \eta\, p(z_{i,t}|\boldsymbol{x}_{i,t})\overline{bel}(\boldsymbol{x}_{i,t}) \end{cases}, \tag{9}$$

where η is a normalization constant[8]. The belief $bel(\boldsymbol{x}_{i,0})$ at $t=0$ is initialized with a uniform distribution. We randomly and uniformly deploy N particles in the image plane at the beginning.

In order to reduce the computational cost, we just consider the position information in the state variable. We adopt a Multi-variate Gaussian transition model in this paper. The positions at time t can be predicted using a Gaussian distribution centred at the transformed positions. The transition probability is described as follows,

$$\begin{aligned} &p(\boldsymbol{x}_{i,t}|\boldsymbol{u}_{i,t},\boldsymbol{x}_{i,t-1}) = \\ &\frac{1}{\sqrt{(2\pi)^k|\Sigma|}}\exp\big[-\frac{1}{2}(\boldsymbol{x}_{i,t}-\boldsymbol{\mu}_{i,t})^T)\Sigma^{-1}(\boldsymbol{x}_{i,t}-\boldsymbol{\mu}_{i,t})\big], \end{aligned} \tag{10}$$

where $\boldsymbol{\mu}_{i,t}$ is the ego-motion compensated version of $\boldsymbol{x}_{i,t-1}$, $|\Sigma|$ is the determinant of the variance Σ, k is the dimension of the problem. We delete the particles that are out of the image range after the motion update.

The measurement probabilities exist in the weights of the particles. As mentioned before, the intensity value of a pixel in the differencing image encodes the probability of the pixel being the foreground. However, in order to improve the robustness for weight computation. We also consider the neighbouring pixels. The weight w_i for particle i is calculated using a 2-D Gaussian kernel. We choose a circle as the neighbouring area that has M number of pixels. The value

of w_i is given by the following equation,

$$w_{i,t} = \sum_{j=1}^{M} I_d(x_j, y_j, t) \frac{1}{\sqrt{(2\pi)}\sigma} \exp\big(-\frac{d^2}{2\sigma}\big), \qquad (11)$$

where x_j, y_j are the coordinates of pixel j, $I_d(x_j, y_j, t)$ is the intensity value, σ is the standard deviation of the Gaussian kernel, $d = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$ is the Euclidean distance between pixel j and particle i. The weight of a particle is actually a weighted average of the neighbouring pixel values in the differencing image. In the re-sampling stage of the particle filter, we apply the Sequential Importance Re-sampling (SIR) technique to generate new particles. The importance is proportional to the weights of the particles.

D. Vector Quantization-based Segmentation

Vector Quantization (VQ) is a kind of lossy data compression method. It can map a vector set into a subset of itself or a set with less elements. The vectors in the original set are represented by a limited number of different vectors. For VQ in our approach, the vector is composed of the pixel values. The distortion error for a vector $\boldsymbol{v} \in \mathbb{R}^{N\times 1}$ with the quantized one $\boldsymbol{v}^* \in \mathbb{R}^{N\times 1}$ is determined as follows,

$$D(\boldsymbol{v}, \boldsymbol{v}^*) = \frac{1}{N}(\boldsymbol{v} - \boldsymbol{v}^*)^T(\boldsymbol{v} - \boldsymbol{v}^*), \qquad (12)$$

where $N = 3$ in our case, note that we use color depth images. To quantize an image is to map all color vectors into different clusters, so the problem of quantization for an image can be solved by clustering pixels using the position and color information. In this paper, we apply the K-means algorithm on depth images[9] for vector quantization. The reason why we use depth images is that moving objects can be more easily retrieved from depth images than RGB images. As shown in Fig.4, we can clearly find that the walking person can be easily extracted from the quantized depth image.

With the quantized depth images, we can segment the moving object using the MAP estimation. Let $p(s_{k,t})$ denote the probability of cluster k being the foreground. To get the segmentation result is to find the cluster that has the maximum foreground probability. The MAP estimation is able to compute the posterior probability using the likelihood from the tracking results. Let $p(m|s_{k,t})$ denote the posterior belief from the particle filter. The posterior probability $p(s_{k,t}|m)$ can be computed by the following equation,

$$p(s_{k,t}|m) = \frac{p(m|s_{k,t})p(s_{k,t})}{p(m)}, \qquad (13)$$

so the segmentation problem becomes to find the cluster k that corresponds to the maximum posterior probability,

$$k = \underset{k}{\operatorname{argmax}}\ p(s_{k,t}|m). \qquad (14)$$

The probability $p(s_{k,t})$ is the prior probability for cluster k being the foreground, $p(m|s_{k,t})$ is the likelihood from the tracking stage, $p(m)$ is a normalization constant. For each segmentation the current depth image is re-quantized, so the prior probability is in a uniform distribution all the time. We use the proportion of particles that lie in the cluster k to model the likelihood. The likelihood $p(m|s_{k,t})$ is given by the following equation,

$$p(m|s_{k,t}) = \frac{n_t}{N_t}, \qquad (15)$$

where n_t is the number of particles that lie in the cluster k at time t, N_t is the total number of the particles at time t. The value of $p(m|s_{k,t})$ encodes the likelihood for each cluster being the foreground. Higher value indicates higher possibility being the foreground.

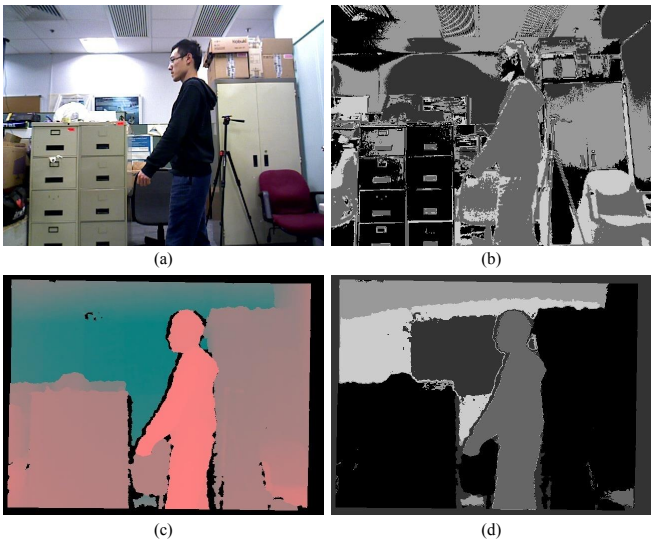

Fig. 4. The comparison between the quantized RGB and depth images. The cluster number is 5. (a) is the original RGB image, (b) is the quantized RGB image, (c) is the original depth image (depth increases from red to green), (d) is the quantized depth image. We can see the person can be more easily identified due to the less texture information in the depth image.

IV. Experimental Results and Discussion

A. Experimental Setup

We perform the experiments using Asus Xtion. Due to the intrinsic limitation of the RGB-D camera, we only perform experiments in indoor environments. The tested indoor environments in this paper are listed as follows,

- Environment I is a common indoor office room. No remarkable negative factor for camera sensing exists.
- Environment II is an office room which contains shiny surfaces such as glass windows or monitor screens. The shiny surfaces can lead to unstable or invalid depth measurements.
- Environment III is a common indoor hallway. There exist large out-of-range depth areas when using some ego-motion patterns.
- Environment IV is similar as the second environment. It has an large paper board for occlusion test.

Walking persons are used as moving objects in this paper, because walking persons are the most common moving objects in indoor environments. The person walks in a normal speed during the experiments. Four typical camera ego-motion patterns are considered. They are listed as follows,

TABLE I

QUANTITATIVE RESULTS FOR OUR EXPERIMENTS IN VARIOUS SCENARIOS (Left : ϕ, Right : π)

Ego-motion Patterns	Environment I		Environment II		Environment III		Environment IV	
Parallel (Mobile Robot)	97.50%	100.00%	92.00%	100.00%	97.78%	86.36%	87.76%	100.00%
Circle (Mobile Robot)	97.30%	100.00%	90.48%	100.00%	93.18%	97.56%	87.88%	98.28%
Opposite (Mobile Robot)	97.56%	97.50%	90.91%	95.00%	97.78%	97.73%	85.71%	95.83%
Irregular (Hand-held)	95.00%	96.49%	84.51%	98.33%	99.03%	98.04%	87.14%	93.44%

- Parallel: The camera moves in parallel with the working person.
- Circle: The camera rotates around a fixed point when the person is walking.
- Opposite: The camera and the person are moving in an opposite direction. They will meet somewhere finally.
- Irregular: The camera is hand-held by a person to realize irregular ego-motion.

The first three ego-motion patterns belong to the regular motion. We realized them by fixing the RGB-D camera on a mobile robot. The last one belongs to the irregular motion. We realized it by hand-held. It should be noted that our approach requires no prior information including the above mentioned ego-motion patterns.

Our program use the VGA resolution. It runs on a laptop with an Intel i3 CPU. The numbers of particles and clusters are fixed to 1000 and 5 respectively. Other parameters, such as the thresholds in RANSAC, are set empirically to moderate values. Our program can be real-time for motion detection with GPU-SURF. The most time-consuming operation is the image segmentation, which costs about 500ms.

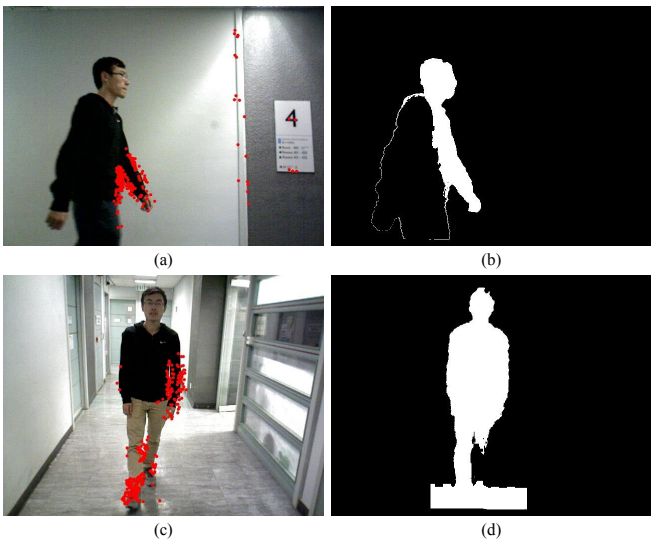

Fig. 5. Two typical motion removal failure cases. (a) and (c) are captured under the *Parallel* and *Irregular* ego-motion cases in environment III. Red dots are the particles which show the tracking results. (b) and (d) are the mask images which show the segmentation results.

B. Evaluation Metrics

The motivation of our approach is to remove moving objects out of the image sequences captured from moving platforms. Considering our potential applications, it is important to check whether our approach is able to fully filter out moving objects, rather than measure the preciseness or recall values of the image segmentation. Thus, we propose an metric the Rate of Successful Motion Removal (RSMR) which is denoted by π to evaluate the motion removal performance. The successful motion removal is confirmed when the moving object can be fully filtered out.

$$\pi = \frac{m}{f}, \tag{16}$$

where m is the number of segmentation results that can fully filter out the moving object, f is the number of successfully tracked frames. Only the successfully tracked frames are used to evaluate π. In order to evaluate the tracking results, we adopt the commonly used evaluation metric the Rate of Successful Tracking (RST) which is denoted by ϕ. The successful tracking is confirmed when the moving object cluster has the maximum number of particles lain on it.

$$\phi = \frac{f}{F}, \tag{17}$$

where f is the number of frames that are successfully tracked, F is the total number of frames in a sequence. The value approximates 40-60 for each test, but 100 for the hand-held case in the hallway.

C. Results and Discussions

Table I summarizes the quantitative results of our experiments. We can see our approach is able to provide practical tracking and segmentation performances. Our approach can achieve 100% success for segmentation especially in the *Parallel* and *Circle* ego-motion cases.

Tracking failures are mainly caused by the false positives of motion detection. The wrongly classified motions can distract and re-distribute the particles. Unfavourable factors such as fast movements, suddenly stopping, sensor noises or object occlusions can lead to the false positives. For example, the shiny surfaces in environments II and IV can cause unstable measurements, so we can see the performances are a little bit degraded. The involvement of partial occlusions can lead to the tracking lost when the person moves in or out of the paper board. Because those unfavourable factors are unavoidable in our experiments, the values of ϕ for all experiments cannot achieve 100%.

Fig.5 shows two typical motion removal failure cases in our experiments. In (a), the person is over-segmented because the number of clusters seems too large for this case. In (c),

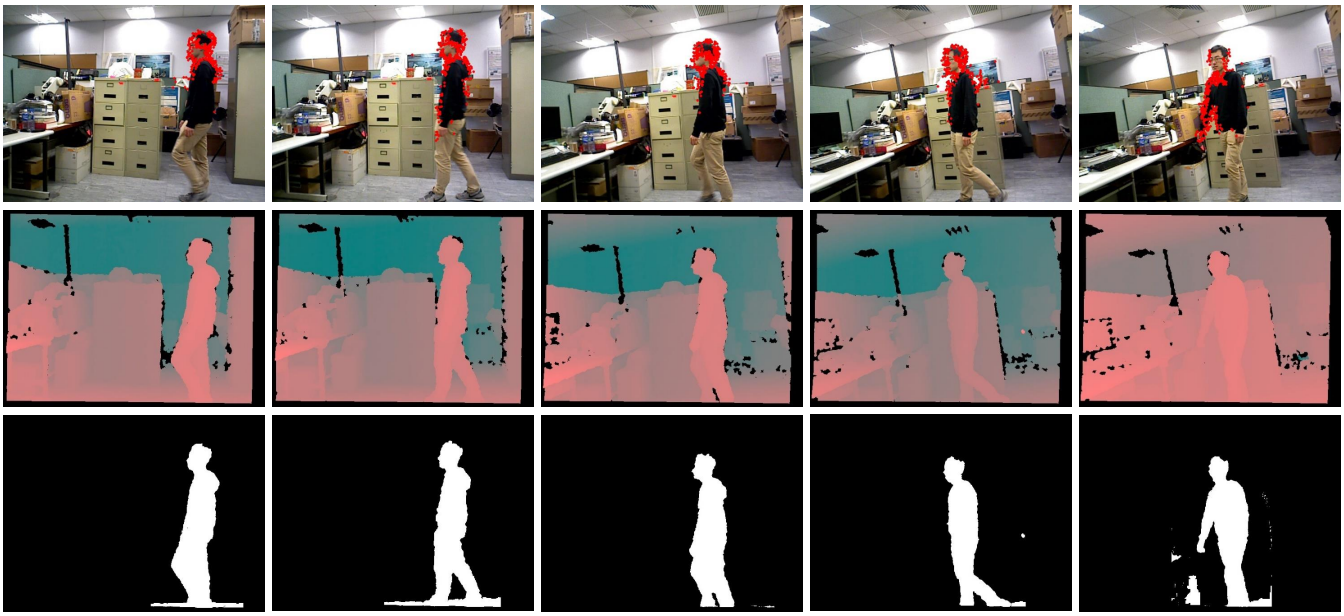

Fig. 6. The successful results in the hand-held case at environment I. The rows from the top to down are the tracking results, the corresponding depth images and the segmentation results. The width of the ROI is 1.4 times of the width of the rectangular contour of the particles.

the distance between the body and the leg is so large that the algorithm fails to classify the whole body into one cluster. There exist some parts of the background that have similar distance as the walking person to the camera. Those parts of background can be mis-classified as foreground due to the similar depth data. To avoid this case, we can do the segmentation within a Region-of-interest (ROI) provided by the rectangular contour of the particles. In Fig.5(c), we set the width of the ROI to 1.3 times of the width of the rectangle. The height of the ROI is set to the height of the image. Note that the quantitative results are obtained without using ROI.

Fig.6 demonstrates sample successful tracking and segmentation results. We can see our approach is able to track and segment the moving object correctly. In the figures, the particles tend to converge on parts of the body that have higher speed. This is because these parts provide higher motion detection likelihood. Due to the unstable depth measurements at object boundaries, the segmentation results cannot be precisely consistent with the moving object.

V. Conclusions

In this paper, we proposed a novel approach for motion removal using a moving RGB-D camera. No prior knowledge is assumed in our approach. Only the visual and depth data from the RGB-D camera are used. The experimental results demonstrate that our approach can provide a practical solution for motion removal on moving platforms. Our approach adopts an online framework and can keep the time cost at a relative low level when using a low-end laptop.

VI. Acknowledgements

Research presented in this paper was partially supported by the Research Grant Council of Hong Kong SAR Government, China, under project No.16206014 and No.16212815; National Natural Science Foundation of China No.6140021318, awarded to Prof. Ming Liu, and partially supported by the RGC projects CUHK#415512 and #CUHK6/CRF/13G awarded to Prof. Max Q.-H. Meng. The authors would like to thank Zhe Min for the experiments preparation.

References

[1] A. Oliver, S. Kang, B. C. Wnsche, and B. MacDonald, "Using the Kinect as a navigation sensor for mobile robotics," in Proceedings of the 27th Conference on Image and Vision Computing New Zealand, 2012, pp. 509-514.

[2] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D Mapping With an RGB-D Camera," Robotics, IEEE Transactions on, vol. 30, pp. 177-187, 2014.

[3] S. W. Kim, K. Yun, K. M. Yi, S. J. Kim, and J. Y. Choi, "Detection of moving objects with a moving camera using non-panoramic background model," Machine vision and applications, vol. 24, pp. 1015-1028, 2013.

[4] B. Jung and G. S. Sukhatme, "Real-time motion tracking from a mobile robot," International Journal of Social Robotics, vol. 2, pp. 63-78, 2010.

[5] W. Liu, S. Yang, and M. Liu, "A 6D-pose estimation method for UAV using known lines," in Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE International Conference on, 2015, pp. 953-958.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24, pp. 381-395, 1981.

[7] Y. Sun, M. Liu, and M. Q.-H. Meng, "WiFi signal strength-based robot indoor localization," in Information and Automation (ICIA), 2014 IEEE International Conference on, 2014, pp. 250-256.

[8] Y. Sun and M. Q. H. Meng, "Multiple moving objects tracking for automated visual surveillance," in Information and Automation, 2015 IEEE International Conference on, 2015, pp. 1617-1621.

[9] M. Liu, F. Pomerleau, F. Colas, and R. Siegwart, "Normal estimation for pointcloud using GPU based sparse tensor voting," in Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on, 2012, pp. 91-96.