Achieving Undelayed Initialization in Monocular SLAM with Generalized Objects Using Velocity Estimate-based Classification

Chen-Han Hsiao and Chieh-Chih Wang

Abstract— **Based on the framework of simultaneous localization and mapping (SLAM), SLAM with generalized objects (GO) has an additional structure to allow motion mode learning of generalized objects, and calculates a joint posterior over the robot, stationary objects and moving objects. While the feasibility of monocular SLAM has been demonstrated and undelayed initialization has been achieved using the inverse depth parametrization, it is still challenging to achieve undelayed initialization in monocular SLAM with GO because of the delay decision of static and moving object classification. In this paper, we propose a simple yet effective static and moving object classification method using the velocity estimates directly from SLAM with GO. Compared to the existing approach in which the observations of a new/unclassified feature can not be used in state estimation, the proposed approach makes the uses of all observations without any delay to estimate the whole state vector of SLAM with GO. Both Monte Carlo simulations and real experimental results demonstrate the accuracy of the proposed classification algorithm and the estimates of monocular SLAM with GO.**

I. INTRODUCTION

The feasibility of monocular simultaneous localization and mapping (SLAM) has been demonstrated [1], [2] in which the 6 degree-of-freedom (DOF) camera pose and 3 DOF feature locations are simultaneously estimated using a single camera following the extended Kalman filter (EKF) framework. An inverse depth parametrization [3], [4] was proposed to accomplish undelayed initialization in monocular SLAM. It is also shown that the inverse depth parametrization has a better Gaussian property for EKF and has a capability for estimating features at a potentially infinite distance. However, SLAM could fail in dynamic environments if moving entities are not dealt with properly [5].

A few attempts have been made to solve the monocular or visual SLAM problems in dynamic environments. Sola [6] mentioned the observability issue of bearings-only tracking and pointed out the observability issues of monocular SLAM in dynamic environments. Accordingly, two cameras instead of single one were used to solve the monocular SLAM and moving object tracking problem with some heuristics for detecting moving objects. In [7], moving object effects are removed given that the geometry of known 3D moving objects are available in which manual operations such as deleting features on non-static objects are needed. Migliore *et al.* [8] demonstrated a monocular simultaneous localization, mapping and moving object tracking (SLAMMOT) system where a SLAM filter and a moving object tracking filter are used separately. The stationary and moving object classification is based on the uncertain projective geometry [9].

In our ongoing work following the SLAM with GO framework [5], an augmented state SLAM with GO approach using the existing inverse depth parametrization was proposed to solve monocular SLAM and bearings-only tracking concurrently. A stereo-based SLAM with GO system has been demonstrated [10] in which the observability issue is solved straightforwardly. In these monocular- and stereo-based approaches, stationary or moving classification of a new detected feature is accomplished by comparing two local monocular SLAM results: one is local monocular SLAM without adding this new feature and the other is local monocular SLAM under the assumption that this new feature is stationary. The difference of these two hypotheses is temporally integrated using a binary Bayes filter. A threshold is determined for stationary or moving object classification after a fixed number of updates. Although the modified inverse depth parametrization is used, initialization of these monocular and stereo-based systems is still delayed as the observations of new features during the classification stage are not used in the estimation.

In this paper, we propose a simple yet effective static and moving object classification method using the velocity estimates directly from SLAM with GO. A new feature is classified as stationary or moving using two thresholds which are determined intuitively. The number of the time steps for classification is not fixed in the proposed classification method which avoids misclassification due to insufficient observation updates and reduces unnecessary computational cost in the cases that new features can be easily classified. The proposed approach makes the uses of all observations without any delay to estimate the whole state vector of SLAM with GO. Both Monte Carlo simulations and real experimental results demonstrate the accuracy of the proposed classification algorithm and the estimates of monocular SLAM with GO.

II. THEORETICAL FOUNDATION

In this section, the theoretical foundation of monocular SLAM with GO is described.

This work was partially supported by grants from Taiwan NSC (#99-2221-E-002-192) and Intel-NTU Connected Context Computing Center.

C.-H. Hsiao was with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. He is currently with MSI.

C.-C. Wang is with the Department of Computer Science and Information Engineering and the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan e-mail: bobwang@ntu.edu.tw

A. Representation in Monocular SLAM with GO

Following the EKF-based SLAM framework, a state vector χ in SLAM with GO consists of the pose and velocity of the camera/robot and locations and velocities of generalized objects.

$$\chi = (\mathbf{x}_k^\top, \mathbf{o}_k^{1\top}, \mathbf{o}_k^{2\top}, \dots, \mathbf{o}_k^{n\top})^\top \tag{1}$$

where $\mathbf{x}_k$ composes of the camera position $\mathbf{r}^W$ in the world coordinate system, the quaternion defining the camera orientation $\mathbf{q}^W$ in the world coordinate system, and the camera velocity $\mathbf{v}^W$ in the world coordinate system and the camera angular velocity ω^C in the camera coordinate system.

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{r}^W \\ \mathbf{q}^W \\ \mathbf{v}^W \\ \omega^C \end{pmatrix} \tag{2}$$

$\mathbf{o}_k^i$ denotes the i-th generalized object which can be stationary or moving in the framework of SLAM with GO. The existing parametrization for only stationary objects is insufficient to represent generalized objects. In [10], the inverse depth parametrization is added with 3-axis speeds in the world coordinate system to represent generalized objects. Each generalized object is coded with the 9-dimension state vector.

$$\mathbf{o}_k^i = \begin{pmatrix} o_k^{i\top} & v_k^{i\top} \end{pmatrix}^\top \tag{3}$$

$$o_k = \begin{pmatrix} x_k & y_k & z_k & \theta_k & \phi_k & \rho_k \end{pmatrix}^\top \tag{4}$$

where o_k is the 3D location of a generalized object presented using the inverse depth parametrization. $\begin{pmatrix} x_k & y_k & z_k \end{pmatrix}$ is the camera location when this object is first observed.

$$v_k = \begin{pmatrix} v_k^x & v_k^y & v_k^z \end{pmatrix}^\top \tag{5}$$

where v_k denotes the 3-axis velocities of the generalized object in the world coordinate system.

The 3D location of a generalized object w.r.t. to the world coordinate system can be computed as:

$$loc(o_k^i) = \begin{pmatrix} x_k \\ y_k \\ z_k \end{pmatrix} + \frac{1}{\rho_k} \times \mathcal{G}(\theta_k, \phi_k) \tag{6}$$

where the direction vector $\mathcal{G}(\theta, \phi)$ defines the direction of the ray and ρ is the inverse depth between the feature and camera optical center.

B. EKF-based SLAM with GO

In the prediction stage of the EKF algorithm, the constant velocity and constant angular velocity motion model [3] is applied for predicting the camera pose at the next time step as the camera is the only sensor used to accomplish SLAM with GO in this work.

For generalized objects, the constant velocity model is applied and the location of a generalized object at the next frame can be calculated in a closed form:

$$\begin{aligned} o_{k+1}^i &= loc(o_k^i) + v_k^i \cdot \Delta t \\ &= \mathbf{r}^i + \frac{1}{\rho_k}\mathcal{G}_k + v_k^i \cdot \Delta t \\ &= \mathbf{r}^i + \frac{1}{\rho_{k+1}}\mathcal{G}_{k+1} \end{aligned} \tag{7}$$

where $\mathcal{G}_k$ and $\mathcal{G}_{k+1}$ are the directional vectors.

In the observation update stage of EKF, generalized objects are transformed to the camera coordinate system and then projected on the camera image plane. Let R_k^c be the rotation matrix defined by the camera orientation $\mathbf{q}_k^c$. The points are transformed to the camera coordinate by:

$$\mathbf{h}_k^{o_i} = \begin{pmatrix} h_x^{o_i} \\ h_y^{o_i} \\ h_z^{o_i} \end{pmatrix} = h(\mathbf{o}_k^i, \mathbf{x}_k) \tag{8}$$

$$= \mathrm{R}_k^c \left(\mathbf{r_k^i} + \frac{1}{\rho_k}\mathcal{G}_k(\theta_k, \phi_k) - \mathbf{r}_k^W \right) \tag{9}$$

The predicted measurements on the image plane are:

$$\mathbf{z}_k^{o_i} = \begin{pmatrix} u \\ v \end{pmatrix} = Proj(\mathbf{h}_k^{o_i}) = \begin{pmatrix} u_0 - \frac{f}{d_x}\frac{h_x^{o_i}}{h_z^{o_i}} \\ v_0 - \frac{f}{d_y}\frac{h_y^{o_i}}{h_z^{o_i}} \end{pmatrix} \tag{10}$$

where $Proj$ is the project function, (u_0, v_0) is the camera center in pixels, f is the focal length, d_x and d_y represent the pixel size. The monocular SLAM with GO state vector is updated by the EKF algorithm.

C. Undelayed Initialization

The location initialization of a new generalized object is the same as the approach in [3]. To initialize the velocity of a generalized object, $\mathbf{v}_0$ is set to be 0 in this work. The covariance value of the velocity estimate $\sigma_\mathbf{v}$ is designed to cover its 95% acceptance region $[-|\mathbf{v}|_{max}, |\mathbf{v}|_{max}]$.

$$\sigma_\mathbf{v} = \frac{|\mathbf{v}|_{max}}{2} \tag{11}$$

$|\mathbf{v}|_{max}$ is set to 3 m/sec in this work. A generalized object is augmented to the state vector of SLAM with GO straightforwardly and the new covariance of the SLAM with GO state is updated accordingly.

By using the modified inverse depth parametrization, a new generalized object is augmented to the state vector at the first observed frame. Through this undelayed initialization, the proposed monocular SLAM with GO system uses all measurements to estimate both the camera pose and the locations generalized objects.

D. Classification in SLAM with GO

In SLAM with GO, generalized objects are represented instead of representing stationary and moving objects directly. From a theoretical point of view, it could be unnecessary to have a stationary and moving object classification module in

SLAM with GO. However, from a practical point of view, stationary objects could further improve the accuracy and convergence of the state estimates under the static object model. After classification, generalized objects can be easily simplified to stationary objects for reducing the size of the state vector or can be maintained as moving objects using the same parametrization of generalized objects. In addition, stationary objects could contribute to loop detection more than moving objects in most cases. Stationary and moving object classification still plays a key role in SLAM with GO.

III. Stationary and Moving Object Classification using Velocity Estimates from monocular SLAM with GO

Stationary and moving object classification from a moving camera is a daunting task. In this section, the proposed classification approach using velocity estimates from monocular SLAM with GO is described in detail.

A. Velocity Convergency

For using velocity estimates from SLAM with GO to classify stationary and moving objects, it should be demonstrated that velocity estimates can be converged and are sufficient to differentiate stationary and moving objects. A simulation using the proposed monocular SLAM with GO is discussed to show the feasibility of the proposed classification algorithm. In this simulated scene, there were 40 static landmarks and 2 moving landmarks. 39 static landmarks were added to the state vector using the inverse depth parametrization as known features. One static landmark (Target 1) and two moving landmarks (Target 2 and Target 3) were initialized as generalized objects at the first observed frame using the modified inverse depth parametrization and were added to the state vector. The camera trajectory was designed as a helix. Fig. 1 shows the velocity estimates of Targets 1, 2 and 3 after 150 EKF updates.

In this simulation, the estimates using the modified inverse depth parametrization are converged to the true values in terms of both locations and velocities whether the target is moving or stationary. The simulation result shows that moving object classification using velocity estimates from SLAM with GO should be feasible.

B. Thresholding Classification

As velocity estimates directly show the motion properties of generalized objects, two simple score functions are defined here for thresholding classification.

1) Score function for detecting static objects: Given the 3-dimension velocity distribution $X = \mathcal{N}(\mu, \Sigma)$ of a generalized object, the score function is defined as:

$$C_s(X) = f_X(0) = \frac{1}{(2\pi)^{3/2}|\Sigma|^{1/2e^{-\frac{1}{2}(0-\mu)^\top\Sigma^{-1}(0-\mu)}}} \quad (12)$$

where f_X is the probability density function of Gaussian distribution X. This score function calculates the probability density function value of the velocity distribution at $(0,0,0)^\top$. The score reveals the relative likelihood of the velocity variable to occur at $(0,0,0)^\top$.

If a generalized object $\mathbf{o}_k^i$ is static, its velocity v_k^i is expected to converge closed to $(0,0,0)^\top$. This score would thus increases and exceeds a threshold t_s.

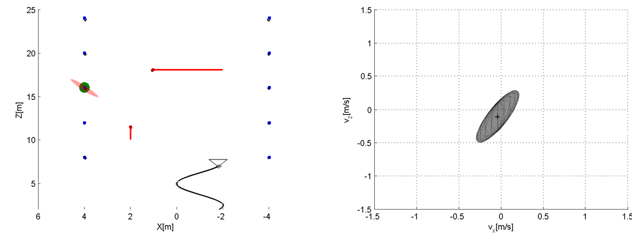
2) Score function for detecting moving objects: Given a 3-dimension velocity distribution $X = \mathcal{N}(\mu, \Sigma)$ of a generalized object, the score function is defined as:

$$C_m(X) = D_X(0) = \sqrt[2]{(0-\mu)^\top\Sigma^{-1}(0-\mu)} \quad (13)$$

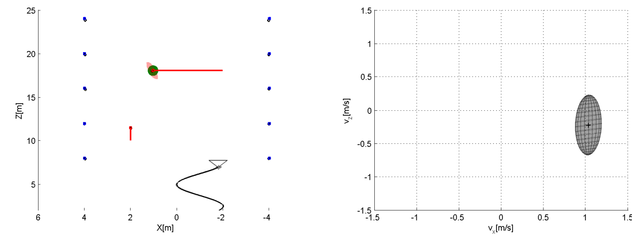
where D_X is the Mahalanobis distance function under distribution X. Mahalanobis distance is often used for data association and outlier detection. For a moving feature $\mathbf{o}_k^i$, its velocity v_k^i is expected to converge away from $(0,0,0)^\top$. The score would thus increases and exceeds threshold t_m if the generalized object is moving.

C. Classification States in SLAM with GO

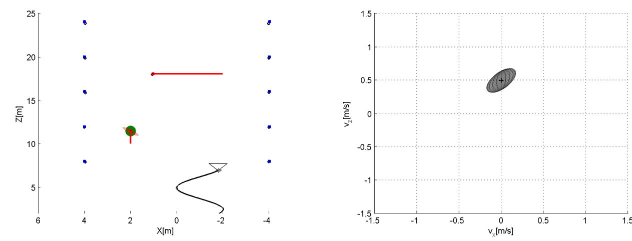
There are three classification states in SLAM with GO: unknown, stationary and moving. Each new feature is initialized at the first observed frame using the modified inverse depth parametrization and the classification state is set to



(a) Target 1 (a static object marked with green circle). The true velocity of Target 1 is $(0,0,0)$

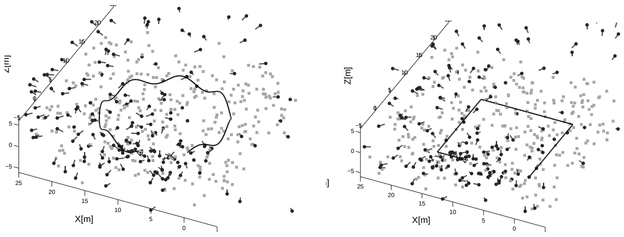


(b) Target 2 (a moving object marked with green circle). The true velocity of Target 2 is $(1,0,0)$



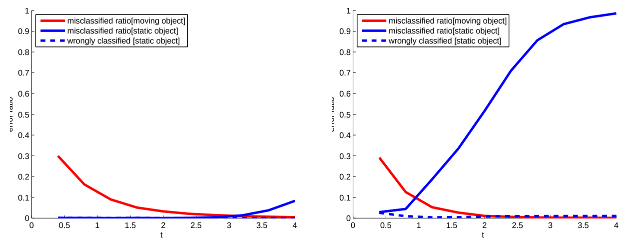
(c) Target 3 (a moving object marked with green circle). The true velocity of Target 3 is $(0,0,0.5)$

Fig. 1. Velocity convergency of 3 targets under an observable condition. The estimates using the modified inverse depth parametrization are converged to the true values.



(a) Observable scenario (b) Unobservable scenario

Fig. 2. The simulation scenarios to evaluate the effects of different thresholds on the classification results. Grey dots denote stationary objects and black dots with lines denote moving objects and their trajectories.



(a) Misclassified ratio in the observable scenario (b) Misclassified ratio in the unobservable scenario

Fig. 3. Effects of the different t_s on the classification results. t_m is fixed at 3.5830.

unknown. In each of the following observed frames, the two score functions are computed based on the estimated velocity distribution for determining if the feature is stationary or moving.

1) From Unknown to Stationary: If the score value $C_s(X)$ of a new feature exceeds the predetermined threshold t_s at a certain frame, this new feature is immediately classified as stationary. The velocity distribution of this feature is adjusted to satisfy the property of a static object in which the velocity is set to $(0,0,0)^\top$ and the corresponding covariance is also set to $\mathbf{0}$. There will be no motion prediction at the prediction stage to ensure the velocity of this object fixed at $(0,0,0)^\top$.

2) From Unknown to Moving: If the score value $C_m(X)$ of a new feature exceeds the predetermined threshold t_m at a certain frame, this feature is immediately classified as moving. As the feature has been initialized with the modified inverse depth parametrization and both the position and the velocity are already being estimated, there is no need to adjust the state distribution and the motion model.

D. Simulation Results

The proposed classification approach is first evaluated using Monte Carlo simulations with the perfect ground truth in this section. The effects of different thresholds and moving object speeds on classification are shown and discussed.

1) Effect of Different Thresholds under Observable and Unobservable Conditions: While the observability issues of SLAM and bearings-only tracking are well understood, SLAM with GO also inherits the observability issues of SLAM and bearings-only tracking. In other words, velocity estimates of generalized or moving objects may not be converged in unobservable conditions. Two scenarios, one is under an observable condition and the other is under an unobservable condition, were designed as depicted in Fig. 2(a) and Fig. 2(b). In Fig. 2(a), the camera moved at a non-constant speed on a circle to avoid unobservable situations. In Fig. 2(b), the camera moved at a constant speed on four connected lines to test the classification performance under an observable situation. 300 static landmarks and 288 moving landmarks with different speed were randomly located in a 3D cube with a width of 30 meters in each scenario. Each scenario has 50 Monte Carlo simulations.

As there are three possible states (unknown, static and moving), the wrongly classified error and the misclassified error are accordingly defined:

Wrongly Classified Error: If a feature is classified as a different type as it should be, this feature is wrongly classified. For instance, a static feature is classified as moving or a moving feature is classified as stationary.

Misclassified Error: If a feature is wrongly classified or not be classified as either static or moving, this feature is misclassified.

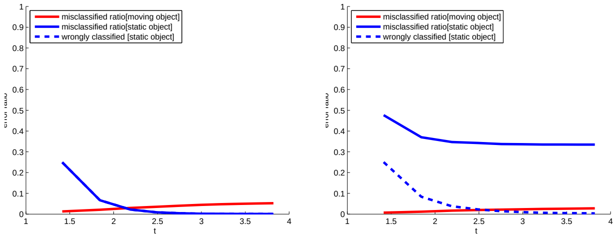
a) Effects of t_s: Fig. 3(a) and Fig. 3(b) show the effects of t_s under an observable and an unobservable condition, respectively. In both scenarios, the misclassified ratio of stationary features increases when the threshold t_s increases, while the misclassified ratio of moving features decreases.

The misclassified ratio of moving features is decreasing when t_s is increasing under the observable situation. Meanwhile, the misclassified ratio of moving features is decreasing when t_s is increasing under the unobservable situation.

These results satisfy the expectation that a larger threshold t_s would result in less features classified as static. Thus, when a larger threshold t_s is chosen, the misclassified ratio of static features would increase and misclassified ratio of moving features would decrease. The trade-off between these two misclassified ratios could be considered according to the usage of the monocular SLAM with GO system.

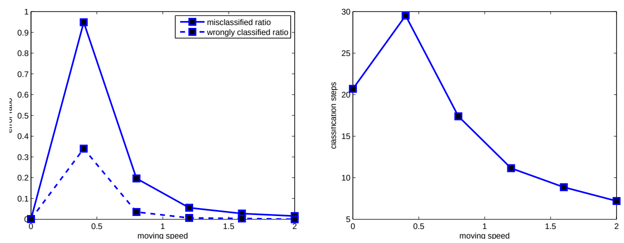
Furthermore, the classification performance is better under observable situations than under unobservable conditions by comparing Fig. 3(a) and Fig.3(b). The observable situations could be critical to achieve better classification performance based on the proposed approach.

b) Effects of t_m: Fig. 4(a) and Fig. 4(b) show the effects of t_m. In both scenarios, the misclassified ratio of static features decreases when the threshold t_m increases, while the misclassified ratio of moving features increases. This finding satisfies our expectation that a larger threshold t_m would result in less features classified as moving. Accordingly, when a larger threshold t_m is chosen, the misclassified ratio of static features would decrease and misclassified ratio of moving features would increase. However, it should be noted that only a small portion of misclassified features are caused by wrongly classification. This means that the proposed classification algorithm does not provide incorrect results. The situations should be more about insufficient data for classification.



(a) Misclassified ratio in the observable scenario (b) Misclassified ratio in the unobservable scenario

Fig. 4. Effect of different t_m on the classification results. t_s is fixed at 1.6.



(a) Classification performance on moving objects with different speeds (b) Number of frame needed for classification

Fig. 5. Effects of speed variation of moving objects on classification.

2) *Effects of Speed Variation of Moving Objects:* In this simulation, the effects of speed variation of moving objects are evaluated. Fig. 5(a) shows that the classification error decreases when the speeds of moving objects increase under the setting of $t_s = 1.6$ and $t_m = 3.5830$. Regarding stationary objects, the error ratio is near 0 which means that almost all stationary objects are correctly detected.

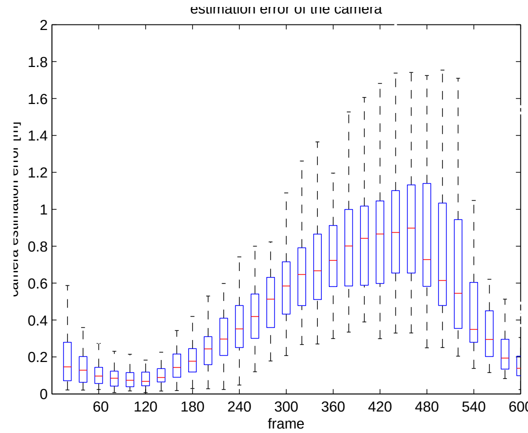
Fig. 5(b) shows the number of frames needed for classification. Recall that the number of frames for classification is data-driven and not fixed in the proposed approach. The result shows that the number of frames needed decreases when the speeds of moving objects increase.

These two findings satisfy the expectation that moving objects at higher speed can be detected within fewer frames and the classification results are more accurate.

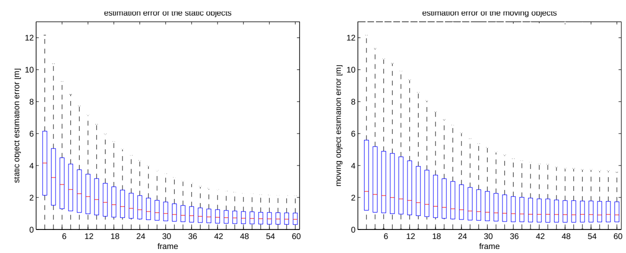
3) *Convergency of our SLAM algorithm:* The convergency of the proposed monocular SLAM with GO algorithm in the observable scenario is checked. The estimate errors of the camera, static features and moving features of 50 Monte Carlo simulation results are shown in Fig. 6. The errors of the camera pose estimates increase when the robot is exploring the environment from Frame 1 to Frame 450. The camera starts to close loop from Frame 450 and the errors decrease which is depicted in Fig. 6(a). Fig. 6(b) and Fig. 6(c) show that the estimate errors of static features and moving features decrease when the number of the frames increases.

IV. Classification Failure in Unobservable Situations

In this section, we discuss the failures of the proposed stationary and moving object classification in unobservable situations.



(a) The estimate errors of the camera pose. Exploring: Frame 1 to Frame 450), Revisiting: Frame 450 to Frame 600.



(b) The estimate errors of the static objects (c) The estimate errors of the moving objects

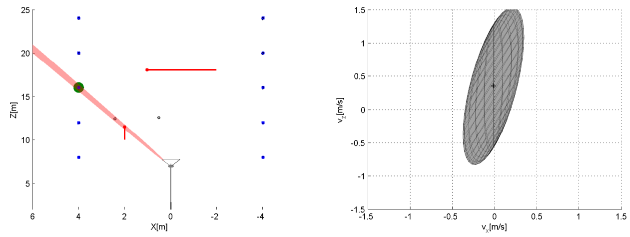
Fig. 6. Convergency of the proposed SLAM with GO algorithm shown with boxplot. The lower quartile, median, and upper quartile values of each box shows the distribution of the estimate errors of all the objects in each observed frame.

A. Unobservable Situations

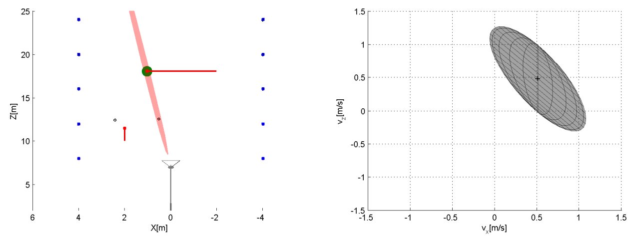
As shown in the previous section, the proposed classification approach could be less reliable in unobservable situations than in observable situations. A simulation was designed to analyze the effects of unobservable situations. The scenario here is the same as the scenario in Fig. 1 except that the camera moves at a constant velocity. Fig. 7 shows the velocity distribution of these 3 targets using the modified inverse depth parametrization after 150 EKF steps.

The results show that the estimate uncertainties of these targets could be too large to reliably determine if these targets are static or moving. In addition, the observations of Target 1 are the same as Target 3 under this camera trajectory even though Target 1 is static and Target 3 is moving. It is impossible to correctly classify Target 1 and Target 3 using the observations from a camera moving at a constant velocity. This means that we can find a corresponding moving object whose observations are the same as a specific stationary object in unobservable situations. Note that such corresponding moving objects must moves parallelly to the camera and at some specific speeds.

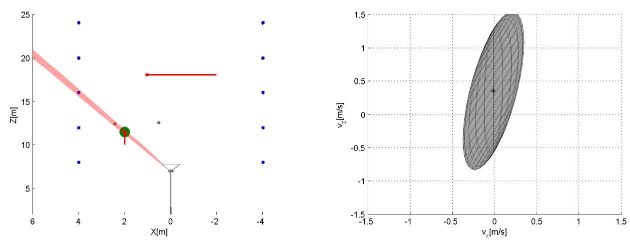
However, the velocity distribution of Target 2 reveals another fact. The 95% confidence region of the velocity estimate of Target 2 does not cover the origin point $(0, 0, 0)^\top$. This means that Target 2 can be correctly classified as moving using the proposed approach even in unobservable



(a) Target 1 (static object marked with green circle). The true velocity of Target 1 is $(0, 0, 0)$



(b) Target 2 (moving object marked with green circle). The true velocity of Target 2 is $(1, 0, 0)$



(c) Target 3 (moving object marked with green circle). The true velocity of Target 3 is $(0, 0, 0.5)$

Fig. 7. Velocity convergency of 3 targets in an unobservable condition.

situations. We argue that no static object would have the same projection as the non-parallelly moving objects. Therefore, it is feasible to determine thresholds to correctly classify non-parallelly moving objects as moving under unobservable situations.

V. Experimental Results

Fig. 8 shows the robotic platform, NTU-PAL7, in which a Point Grey Dragonfly2 wide-angle camera was used to collect image data with 13 frames per second, and a SICK LMS-100 laser scanner was used for ground truthing. The field of view of the camera is 79.48 degree. The resolution of the images is 640×480. The experiment was conducted in the basement of our department. 1793 images were collected for evaluating the overall performance of monocular SLAM with GO such as loop closing, classification and tracking. In this data set, there is a person moving around and appearing 3 times in front of the camera.

There are 107 static features and 12 moving features. When the person appeared, 4 features on the person were generated and initialized. Table I shows the performance of the proposed classification algorithm. None of the feature is wrongly classified. 107 static features in the environment are all classified correctly as static, and 12 moving features are also classified correctly as moving.

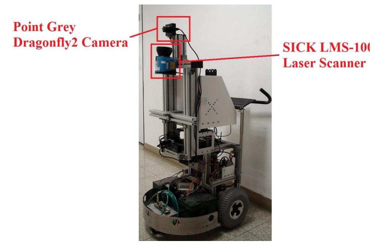


Fig. 8. The NTU-PAL7 robot.

	classification state		
	static	moving	unknown
Static	107	0	0
Moving	0	12	0

TABLE I

Total classification result of real experiment

Fig. 9 shows some of the input images and corresponding SLAM with GO results. At the beginning of the experiment, the checkerboard with known sizes was used for estimating the scale. It is demonstrated that the camera poses are properly estimated, the 3D feature-based map is constructed using the proposed monocular SLAM with GO algorithm, and moving features are correctly detected using the proposed velocity estimate-based classification approach.

VI. Conclusion and Future Work

We proposed a simple yet effective static and moving object classification method using the velocity estimates directly from monocular SLAM with GO. The promising results of Monte Carlo simulations and real experiments have demonstrated the feasibility of the proposed approach. The modified inverse depth parametrization and the proposed classification method achieves undelayed initialization in monocular SLAM with GO. We also showed the interesting issues of classification in unobservable situations.

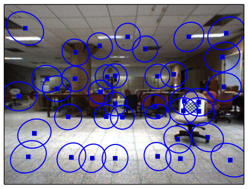
The constant acceleration model and other more advanced motion models should be applied for tracking moving objects with high-degree motion patterns. Evaluating the tracking performance of moving objects with more complicated motion pattern using SLAM with GO is of our interest. In addition, solutions to move-stop-move maneuvers should be developed.

References

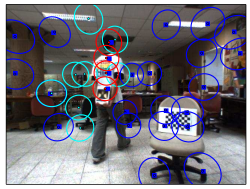
[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[2] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix, "Vision-based slam: Stereo and monocular approaches," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 343–364, September 2007.

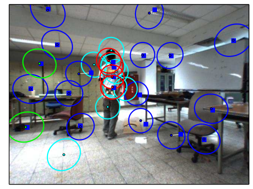
[3] J. M. M. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular slam," in *Robotics: Science and Systems*, Philadelphia, USA, August 2006.



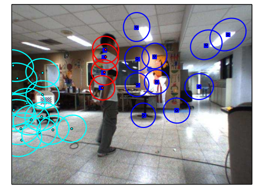
(a) Frame 10.



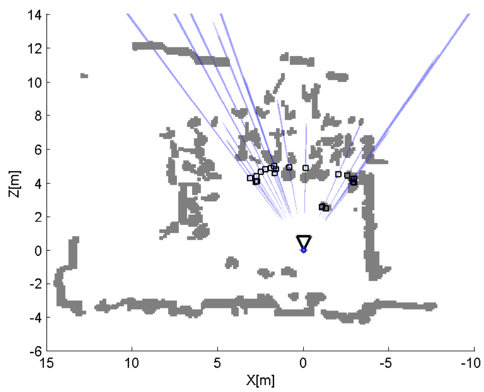
(b) Frame 330. The person appeared at the first time. 4 feature are located and initialized in the state vector.



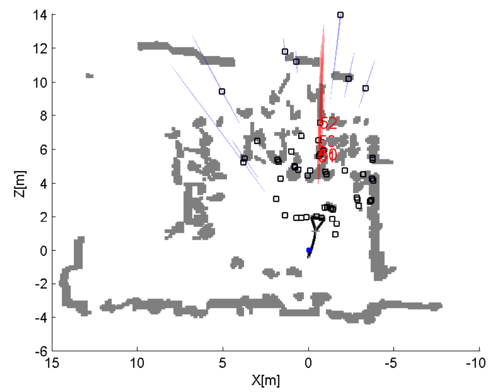
(c) Frame 950. The person appeared at the second time.



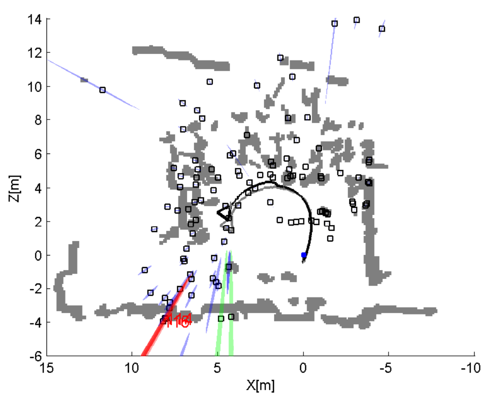
(d) Frame 1350. The person appeared at the third time.



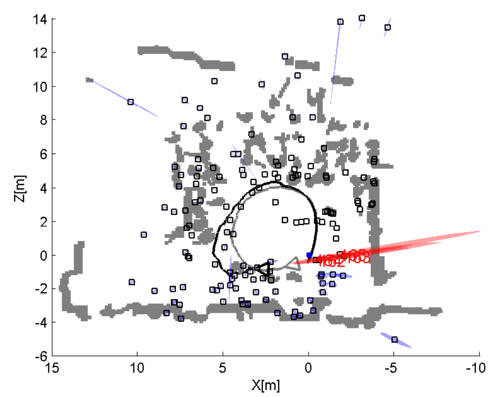
(e) Top view: Frame 10.



(f) Top view: Frame 330.



(g) Top view: Frame 950.



(h) Top view: Frame 1350.

Fig. 9. (a)-(d) show the examples of input images and the results of feature extraction and data assoication. (e)-(f) are the top views of the corresponding SLAM with GO results. In (a)-(d), blue squares are static features, red dots are moving features, green dots are new initialized features with unknown states, and cyan dots are non-associated features. Ellipses show the projected 2σ bounds of the features. In (e)-(f), black and grey triangles and lines indicate the camera poses and trajectories from monocular SLAM with GO and LIDAR-based SLAM with DATMO. Gray points show the occupancy grid map from the SLAM part of LIDAR-based SLAM with DATMO. All the estimation of visual features are inside the reasonable cube. Squares indicate the stationary features and blue shadows indicates the 95% acceptance regions of the estimates.

[4] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, October 2008.

[5] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, September 2007.

[6] J. Sola, "Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach." Ph.D. dissertation, Institut National Polytechnique de Toulouse, February 2007. [Online]. Available: http://homepages.laas.fr/jsola/JoanSola/eng/JoanSola.html

[7] S. Wangsiripitak and D. W. Murray, "Avoiding moving outliers in visual slam by tracking moving objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009, pp. 375–380.

[8] D. Migliore, R. Rigamonti, D. Marzorati, M. Matteucci, and D. G. Sorrenti, "Use a single camera for simultaneous localization and mapping with mobile object tracking in dynamic environments," in *ICRA Workshop on Safe navigation in open and dynamic environments: Application to autonomous vehicles*, 2009.

[9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, March 2004.

[10] K.-H. Lin and C.-C. Wang, "Stereo-based simultaneous localization, mapping and moving object tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.