Use a Single Camera for Simultaneous Localization And Mapping with Mobile Object Tracking in dynamic environments

Davide Migliore, Roberto Rigamonti, Daniele Marzorati, Matteo Matteucci, Domenico G. Sorrenti

***Abstract*— The aim of this work is to demonstrate that it is possible to use a single camera to solve the problem of Simultaneous Localization And Mapping in dynamic environments obtaining, at the same time, the estimation of the moving objects trajectories. Specifically, we show that it is possible to segment the features belonging to independently moving objects from a moving camera using a MonoSLAM algorithm together with a Bearing-Only Tracker. The idea is to exchange between two parallel working systems, i.e. the SLAM filter and the bearing-only tracker, information about the pose of the camera and the motion of the feature to improve the robustness of the SLAM algorithm and maintain a consistent estimation of both the pose, the map, and the features trajectories. Experiments in simulated and real environments substantiate that the proposed technique is able to maintain consistent estimations in a fast and robust way suitable for a real-time application, even in situations where classical MonoSLAM algorithms are deemed to fail.**

I. INTRODUCTION

The key prerequisite for a complete autonomous navigation system is a deep understanding of the surrounding world as perceived by robot sensors. In Simultaneous Localization And Mapping (SLAM) literature it is possible to find many solutions using different kind of sensors (i.e. lasers, cameras, sonars) [1], but most of these algorithms assume a static environment or filter out the dynamic elements perceived in the scene [2].

Although the proposed approaches are effective, they are often expensive or complex and not usable for real applications. For this reason, in this paper, we focus on solutions based on a single camera, a small and inexpensive device that allows to have rich information about the environment perceived. In the last years we assisted the proliferation [3] of systems based on a single camera that are able to simultaneously localize themselves in real-time [4], building 3D maps of huge environments [5] and placing virtual elements in the scene [6]. However, as their precursors, they assume again a static environment.

In this paper we want to demonstrate that it is possible to relax the world motionless hypothesis, proposing a method to estimate online the 6 DoF of a camera and the 3D map in presence of generic dynamic objects.

A first remarkable work on this direction was done by Wang et al. [7], who proposed a mathematical framework to solve the problem of Simultaneous Localization And Mapping and Moving Objects Tracking (SLAMMOT), that can be considered the intersection between SLAM and moving object tracking. The authors investigate theoretically the SLAMMOT problem, demonstrating that it is possible to solve it maintaining separate posteriors for stationary and moving objects, and validating the algorithm empirically by analyzing data acquired with a laser rangefinder in real urban environment.

A different approach was presented by Bibby and Reid [8], introducing a technique called SLAMIDE, to combine the least-squares formulation of SLAM and sliding window optimization, together with a generalized expectation maximization method. Their idea is to incorporate both dynamic and stationary objects into SLAM estimation, without splitting the problem in two and considering the possibility of a reversible data association. Simulated experiments demonstrated the capabilities of the proposed solution, which is able to estimate, consistently, the pose and the map also in presence of dynamic features in a unique framework. However, as already demonstrated by Wang [7], the idea of including all the features in the SLAM state reduces the performance of the filter in terms of speed, highlighting the principal drawback of SLAMIDE: the complexity.

A different approach was proposed by Ess et al. [9], who presented a mobile system based on a stereo camera which integrates continuous visual odometry computation with tracking-by-detection, to track pedestrians in spite of frequent occlusions and egomotion of the camera rig. This method obtains interesting results in very challenging scenarios, but it is not a generic solution since it considers only pedestrian/vehicle tracking, and it is not computationally feasible for a robotics application. Moreover, no map is built, since the system is based on a visual odometry, thus it is not possible to have enough information about the environment to allow trajectory planning for an autonomous vehicle.

An approach requiring less computational resources, but still using a stereo camera, was introduced by Solà et al. [10], who described a system based on a framework called BiCamSLAM, that combines the advantages of the monocular reconstruction with the advantages of stereo vision. In his proposal, Solà tries to solve the SLAMMOT problem estimating, at the same time, the position of the robot, the static map and the trajectory of the moving objects. In particular, Solà proposes to separate the SLAM algorithm from the tracking one, adopting a camera-centric representation of the world and using a different filter for each moving object, dropping in this way objects crosscorrelations with

D. Marzorati and D. G. Sorrenti are with Università di Milano - Bicocca, Building U14, v.le Sarca 336, 20126, Milano, Italy{marzorati, sorrenti}@disco.unimib.it

M. Matteucci, R. Rigamonti and D. Migliore are with Politecnico di Milano, via Ponzio 34/5, 20133, Milano, Italy{matteucci, migliore}@elet.polimi.it

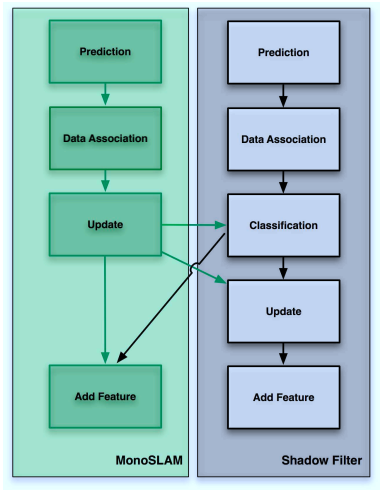


Fig. 1. Schema of the proposed SLAMBOT system. On the left we have the SLAM filter that, as explained in [11], estimates the camera pose and the position of the static elements in the scene by means of the EKF prediction, data association and update steps. The last pose estimated by the SLAM filter is then used by the the Shadow filter to identify dynamic features and estimate their trajectories. Once a feature is classified as static, it is added to the SLAM filter.

the robot's pose.

In this paper, starting from the Solà idea, a viable solution to the online monocular SLAM with moving objects tracking is proposed. The goal of our method is to obtain a consistent map of the static environment, discriminating between static and dynamic objects and being able at the meantime to approximate the trajectories of the moving features.

II. MONOSLAM WITH MOVING OBJECTS

Simultaneous estimation of pose and map based on the analysis of images perceived by a moving observer is not a trivial task; especially when the environment monitored contains dynamic elements that might affect the consistency of the estimates, leading to failure in the traditional SLAM algorithm. In the monocular case, this hindrance is worsened by the reconstruction procedure that is often unable to detect the dynamic behavior of a feature because of the high initial uncertainty associated with it [4].

A possible solution was proposed by Wang et al. [7], under the assumption that moving objects do not carry information about the map and the robot pose: he did not consider them as references for localization because of their inherent instability [10]. Exploiting this insight, we decided to split the estimation process over two filters reciprocally related (see Figure 1): the SLAM filter based on monocular camera (MonoSLAM), that uses static features to estimate map and camera pose, and the tracker, named in this paper "Shadow Filter", that, by knowing the camera pose, deals with the moving features in the environment. The role of the Shadow Filter is twofold: on one side, it tracks the behavior of the moving features, on the other, it retains the new features detected by the camera until it can tag them as static or dynamic, avoiding in this way inconsistencies in the SLAM process.

The system we propose relies on two main assumptions. Since we do not have any odometry measurement (i.e., we do not have an IMU), we need an absolute reference to understand how the camera and the feature are moving. Therefore, before perceiving dynamic features, we initialize the SLAM filter with a set of static features in known position (to estimate the scale), obtaining a first estimation of the camera pose w.r.t. the world frame. Moreover, to ensure consistent estimation and correct features classification during the whole execution of this system, it is important to have in the image and in the SLAM filter state enough static features to maintain an estimation of the absolute reference frame.

Under these assumptions, that could be easily relaxed by the use of an Inertial Measurement Unit (IMU), new features are initialized in the Shadow filter only. To avoid the corruption of the SLAM filter, these features are retained in it until it is not possible to mark them as static, in which case they are passed to the SLAM filter, or dynamic, in which case they are kept in the Shadow filter and tracked along their movements.

The MonoSLAM algorithm used in this work is the same proposed by Marzorati et al. [11], thus we avoid to explain here how this algorithm works, focusing, instead, on the description of the Shadow Filter side of the system and its interaction with the SLAM filter. However, it is simple to notice that the method proposed is independent of the SLAM algorithm used, since the only information exchanged are the camera pose and the feature positions.

III. DYNAMIC FEATURES TRACKING

As explained before, we propose to use a Bearing Only Tracker, the "Shadow Filter", to estimate and classify the new features perceived. Once we know the camera pose from the SLAM filter, to estimate the position and the velocity of a moving feature w.r.t. the camera frame we can use an EKF characterized by the following state:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{F_k}^{C_k} \\ \mathbf{v}_{F_k}^{F_k} \end{bmatrix}, \tag{1}$$

where $\mathbf{x}_{F_k}^{C_k} = (x_f, y_f, z_f, w_f)^T$ are the feature homogeneous coordinates at time k w.r.t. the camera frame C_k and $\mathbf{v}_{F_k}^{F_k} = (v_{f_x}, v_{f_y}, v_{f_z})^T$ is its velocity w.r.t. the feature frame F_k.

At each step we have to maintain the reference of the Shadow filter always w.r.t. the camera frame, thus we need to roto-translate the feature position and rotate the velocity vector before the update step. Assuming constant velocity, we can write the motion equation as:

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{x}_{F_{k+1}}^{C_{k+1}} \\ \mathbf{v}_{F_{k+1}}^{F_{k+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{C_k}^{C_{k+1}} \oplus \mathbf{x}_{F_k}^{C_k} \oplus (\mathbf{v}_{F_{k+1}}^{F_k} \Delta t) \\ \mathbf{x}_{F_k}^{F_{k+1}} \oplus \mathbf{v}_{F_k}^{F_k} \end{bmatrix}, \tag{2}$$

where: $\mathbf{x}_{C_k}^{C_{k+1}}$ is the roto-translation between the camera poses C_k and C_{k+1}, $\mathbf{x}_{F_k}^{C_k}$ is the feature position w.r.t. the camera pose at time k, $\mathbf{v}_{F_{k+1}}^{F_k}$ is the velocity of the feature

at time $k+1$ w.r.t. the feature frame F_k, $\mathbf{v}_{F_k}^{F_k}$ the velocity of the feature at time k w.r.t. the feature frame F_k, $\mathbf{x}_{F_k}^{F_{k+1}}$ is the rotation from the frame reference at time k to the frame reference at time $k+1$ and $\oplus$ is the transformation composition operator. Notice that the state of the feature is somehow represented in a mixed frame of reference to simplify the motion model: its position is in the camera frame, while its velocity is in the feature frame (i.e., the camera reference translated in the feature point).

The measurement equation in homogeneous coordinates can be written as:

$$\mathbf{h}_k = \begin{bmatrix} h_{k_x} \\ h_{k_y} \\ h_{k_z} \end{bmatrix} = \mathbf{M}\mathbf{x}_{F_k}^{C_k}, \tag{3}$$

where $\mathbf{M}$ is the calibration matrix:

$$\mathbf{M} = \begin{bmatrix} f_{c_x} & 0 & cc_x \\ 0 & f_{c_y} & cc_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

and the pixel coordinates on the image plane can be simply obtained as

$$\mathbf{h}_k = \begin{bmatrix} h_{k_u} \\ h_{k_v} \end{bmatrix} \begin{bmatrix} h_{k_x}/h_{k_z} \\ h_{k_y}/h_{k_z} \end{bmatrix}. \tag{5}$$

For the experiments shown in this paper we used a camera with a wide-angle lens, to improve the performance of single-camera SLAM [12], thus the measurement equation should be modified accordingly to take into account the radial distortion, as exposed in [11]. Finally, to estimate iteratively the current position[1] of the feature, we just need to compute the Jacobian of these models and apply the classical steps of the Extedend Kalman Filter.

This approach allows us to have an approximated estimation of the feature pose and, in this way, make inference about its movements.

A. Detecting moving features

To guarantee the correct functioning of the SLAM algorithm, we need to classify new features as dynamic or static before using them to estimate the camera pose and the map. The first time we perceive a feature, we do not know where it is located in the 3D scene, thus we initialize it with a huge uncertainty in the depth. In the next frame, once the feature is associated with a measurement and then updated, its position changes, moving along the projection ray and possibly causing the estimate of false motion. For this reason we can not rely the velocities estimated in the Shadow Filter and we need a more robust classifier.

Referring to the viewing ray as a straight line and to the position from where the feature was viewed the first time, we can make a geometric reasoning, based on an approach that resembles the epipolar constraint. The basic idea is to check continuously the intersections between three viewing rays belonging to the same feature viewed in three different camera poses. If these intersections are not the same during the camera motion or it does not exist, then the feature can be classified as dynamic.

However, in real world, where the moving sensor returns uncertain bearing-only measurements, the previous task is not trivial to solve, since the presence of the uncertainty could affect all the geometric reasoning. To take into account the uncertainty associated with each measurement and each estimate, we need to introduce a probabilistic framework that allows us to check the relationships between the viewing rays in an uncertain world: Uncertain Projective Geometry [15].

Using this framework we can describe, combine, and estimate various types of geometric elements (3D points, 3D lines and 3D planes) maintaing the information about their uncertainty. By the use of Uncertain Projective Geometry, these elements are represented using homogeneous vectors (using the Plücker coordinates for lines) with their covariance matrices, and simple bilinear expressions to represent join and intersection operators are used. This result can be obtained by using two construction matrices: $\mathbf{O}(\cdot)$ (for 3D lines) and $\mathbf{\Pi}(\cdot)$ (for 3D points and 3D planes).

To join two 3D points $\mathbf{X} = (X_1, Y_1, Z_1, W_1)^T$, $\mathbf{Y} = (X_2, Y_2, Z_2, W_2)^T$ into a 3D line $\mathbf{L}$ expressed in Plücker coordinates [15], we can write:

$$\mathbf{L} = \mathbf{X} \wedge \mathbf{Y} = \mathbf{\Pi}(\mathbf{X})\mathbf{Y}, \tag{6}$$

being

$$\mathbf{\Pi}(\mathbf{X}) = \frac{\partial \mathbf{X} \wedge \partial \mathbf{Y}}{\partial \mathbf{Y}} = \begin{pmatrix} W_1 & 0 & 0 & -X_1 \\ 0 & W_1 & 0 & -Y_1 \\ 0 & 0 & W_1 & -Z_1 \\ 0 & -Z_1 & Y_1 & 0 \\ Z_1 & 0 & -X_1 & 0 \\ -Y_1 & X_1 & 0 & 0 \end{pmatrix}. \tag{7}$$

Again we can join a 3D point $\mathbf{X} = (X_1, Y_1, Z_1, W_1)^T$ with a 3D line $\mathbf{L} = (L_1, L_2, L_3, L_4, L_5, L_6)$ into a 3D plane $\mathbf{A}$:

$$\mathbf{A} = \mathbf{X} \wedge \mathbf{L} = \mathbf{O}(\mathbf{L})\mathbf{X}, \tag{8}$$

$$\mathbf{O}(\mathbf{L}) = \frac{\partial \mathbf{X} \wedge \partial \mathbf{L}}{\partial \mathbf{X}} = \begin{pmatrix} 0 & L_3 & -L_2 & -L_4 \\ -L_3 & 0 & L_1 & -L_5 \\ L_2 & -L_1 & 0 & -L_6 \\ L_4 & L_5 & L_6 & 0 \end{pmatrix}. \tag{9}$$

These construction matrices are useful tools to derive new geometric entities from known ones, e.g. a 3D line from two 3D points, a 3D point from the intersection of two 3D lines, etc.; at the same time, being bilinear equations, these operators directly represent the Jacobian of the transformation which is used for the uncertainty propagation in the construction process.

A new entity z can be estimated from two entities x and y, with a simple matrix multiplication:

$$z = f(x, y) = U(x)y = V(y)x, \tag{10}$$

where $U(x)$ and $V(y)$ are, at the same time, the bilinear operators and the Jacobian of the x and y entity respectively.

[1]Notice that this filter can estimate the trajectories of the moving points up to a scale factor [13], however it is possible to overcome this drawback initializing the correct scale in the first frame, as showed in [14].

Assuming the entities to be uncertain, the pairs (x, Σ_{xx}) and (y, Σ_{yy}), and possibly the covariances Σ_{xy} between x and y, are required for computing the error propagation as:

$$(z, \Sigma_{zz}) = \left(U(x)y, [V(y), U(x)] \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{xy} & \Sigma_{yy} \end{pmatrix} \begin{bmatrix} V^T(y) \\ U^T(x) \end{bmatrix} \right), \quad (11)$$

and in case of independence between x and y we obtain:

$$(z, \Sigma_{zz}) = \left(U(x)y, U(x)\Sigma_{yy}U^T(x) + V(y)\Sigma_{xx}V^T(y) \right). \quad (12)$$

To check the geometric relationship between two geometric entities it is then possible to use a statistical test on the distance vector d defined using the previous bilinear equation. In particular a relation can be assumed to hold if the hypothesis

$$H_0 : d = U(x)y = V(y)x = 0 \quad (13)$$

cannot be rejected. Notice that the hypothesis H_0 can be rejected with a significance level of α if

$$T = d^T \Sigma_{dd}^{-1} d > \varepsilon_H = \chi^2_{1-\alpha;n} \quad (14)$$

To perform the test, we need to fix the probability α that we reject H_0 although it is actually true and this situation is called Type-I error. The probability α is usually a small number such as 1% or 5% and it is called significance level of the test. The critical value ε_H such that $P(T > \varepsilon_H | H_0) = \alpha$ is given by the $(1-\alpha)$-quantile of the χ^2 distribution. It is crucial to note that a successful hypothesis test $T < \varepsilon_H$ does not validate that H_0 is true, it merely states that there is not enough evidence to reject H_0.

The covariance matrix Σ_{dd} of d is given by first order error propagation as

$$\Sigma_{dd} = U(x)\Sigma_{yy}U^T(x) + V(y)\Sigma_{xx}V^T(y)$$

In general Σ_{dd} may be singular, if d is a n x 1 vector, r is is the degree of freedom of the relation R and $r < n$. The singularity causes a problem, as we have to invert the covariance matrix. But, at least for projective relations, it can be guaranteed that the rank of Σ_{dd} is not less than r (see Heuel [15] for more details).

IV. EXPERIMENTAL RESULTS

In this section we want to test the capabilities of our system, verifying the result of dynamic classification and the consistence of the estimated position and map. Before trying the algorithm with real data, we verified the consistency of the Shadow filter, testing it in a simulated framework, in which a moving camera was put inside an environment where another dynamic element is moving in the scene (see Figure 2 for a reference). At each time the correct camera position is passed to the Shadow filter and the trajectory of the feature is estimated. As it is possible to notice from Figure 3, the estimate remains consistent during the whole process. The uncertainty associated to the depth coordinate (in the case of the experiment this can be identified with the X coordinate) is higher than the uncertainties associated to the other coordinates, making the Shadow filter estimates

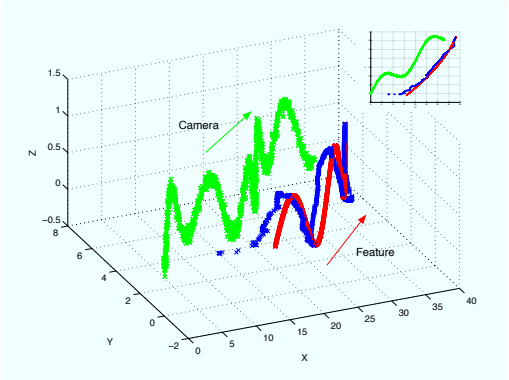


Fig. 2. In this image we show the trajectory of the camera (in green) and of the feature (in red), simulated to test the capabilities of the Shadow filter. In blue it is possible to see the accuracy of the estimated position (the small image represents the projection of the same scene on the XY plane).

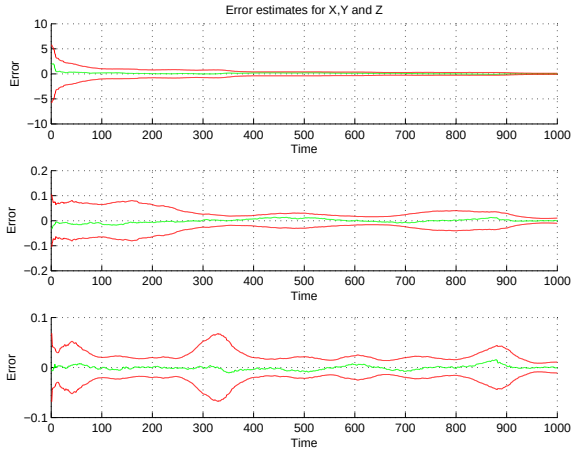


Fig. 3. Consistency test for the Shadow filter. In the plots we show the estimation error (in green) for the x, y, z coordinates of the feature respectively. In red we present the $\pm 3\sigma$ threshold for the covariance; notice that the errors are always contained between those bounds, thus this filter remains consistent.

unfeasible for accurate tracking. This drawback is principally due to consecutive violations of the observability conditions. In fact the displacements between two consecutive steps are so small to cause the partial unobservability of the homogenous part of the feature and a consequent increase of the uncertainty associated to the depth component. This simple analysis gives us information about the quality and the accuracy of the estimates, but also provides an important insight: the observability condition can be easily violated in an online MonoSLAM application.

Although we can not localize accurately the moving object, the consistency of the filter demonstrates the validity of the reasoning based on the uncertainty geometry approach (notice that the errors is always included in the $\pm 3\sigma$ uncertainty interval) and it proves that, taking into account the estimate uncertainty, we can robustly classify a feature as static or dynamic.

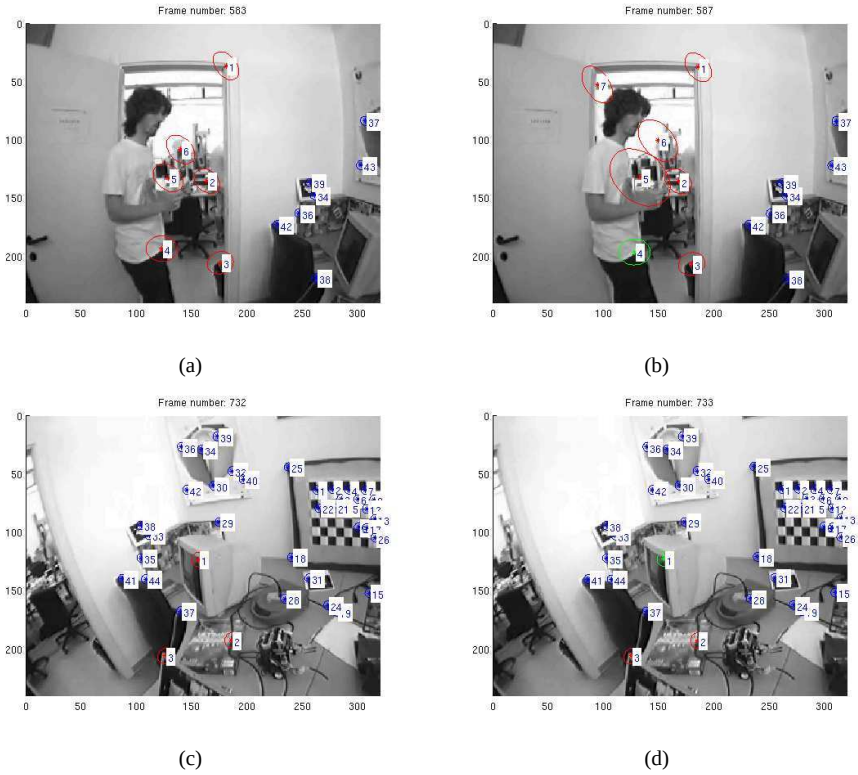


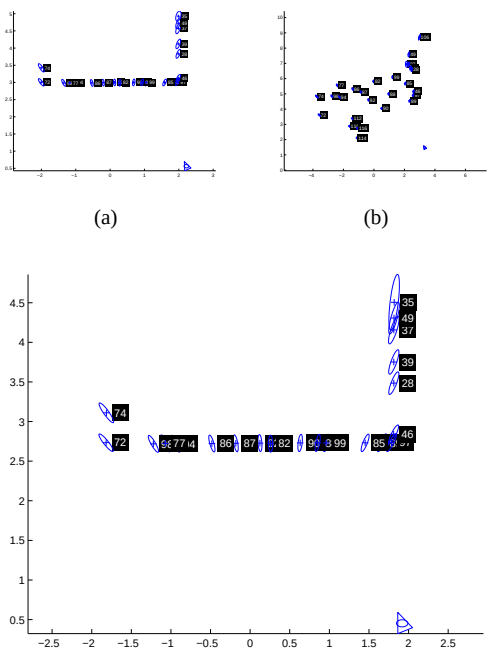
Fig. 4. Static/Dynamic classifier results. In the first row it is possible to see an example of dynamic (in green), static (in blue) classification. The features in the Shadow filter that are waiting for a classification are showed in red. In the last row we can see a feature erroneously classified as moving. This kind of error is expected since the classification is based on a probabilistic test with a threshold of 95%.

This statement can be validated by testing the classifier algorithm on real datasets. In Figure 4 it is possible to see two examples representing both a correct and a wrong classification. We have tested the algorithm using many real datasets and we noticed that, if the feature is correctly matched, the algorithm always distinguish between moving and static features. Sometimes it is possible to have a static feature classified as dynamic (see again Figure 4(c) 4(d)), but it never happened to confuse a moving feature as static. Albeit the probabilistic test has an expected failure rate of the 5%, this contingency happened rarely in our experiments (see again Figure 4(c) 4(d) for an example) and, since it does not corrupt the SLAM filter, it can be tolerated.

Finally we were interested in verifying that our system is able to improve the estimates quality when there are dynamic features in the environment. For this purpose we set up a simulated 3D environment characterized by features both static and dynamic. Data association was performed manually to avoid possible errors due to mismatches and to evaluate the quality of the pose and of the estimated map against a ground truth. In Figure 5 it is possible to see the improvements carried by the use of the Shadow Filter. In the first plot (Figure 5(a)) it is possible to see the map resulting from the use of the classic MonoSLAM algorithm using only the static features. In Figure 5(b) it is shown the results using always the classic MonoSLAM, but this time introducing the dynamic elements, and in the last image (Figure 5(c)) the resulting map obtained introducing the Shadow filter. It is also possible to see how a traditional SLAM filter, that does not identify and exclude from estimation the dynamic features, introduces a set of errors that lead to failure. If we correctly identify the dynamic features, we can avoid to initialize them inside the SLAM filter, maintaining the same accuracy of a SLAM system operating in a purely-static environment. In Figure 6 it is possible to see the results obtained using the real dataset. Despite the presence of dynamic features that could affect the SLAM algorithm, the estimated map remains consistent and, when the camera perceives again the checkerboard, the features are re-matched correctly, closing the loop.

V. CONCLUSIONS AND FUTURE WORKS

In this paper we have proposed a novel solution for the problem of Simultaneous Localization, Mapping and Moving Object Tracking, when using a single camera as a sensor. To avoid errors in the SLAM estimates, we demonstrated that it is possible to identify online the static and dynamic features, using an approach based on the Uncertain Geometry proposed by Heuel [15], that allows to detect the moving features with a simple statistical test. The experimental results confirmed the capabilities of this approach that can be used online in real application and, potentially, with any MonoSLAM algorithm with performances that allow online execution, since it does not require any particular modification of the original SLAM algorithm.



(a) (b)

(c)

Fig. 5. In this image we show the estimated map when we have an environment containing moving feature, using the MonoSLAM proposed in [11] (b) and using the MonoSLAMBOT approach (c). This result can be compared with the map obtained using the MonoSLAM and "disabling" the dynamic features (a).

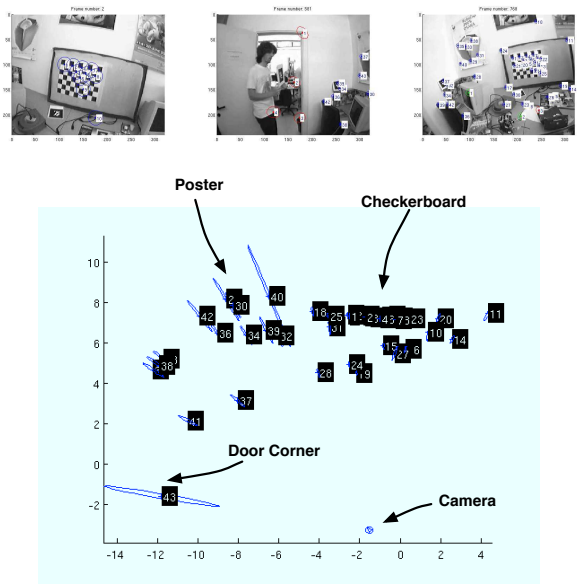


Fig. 6. The resulting map obtained using the MonoSLAMBOT approach is robust to moving elements in the scene (in this case a person). The estimates are consistent and allow the loop closure on the checkerboard.

One limitation of this work is due to the difficulty of tracking robustly the moving elements between frames at different time (e.g., the interest points detected on a walking person, as in Figure 6, could change considerably during the acquisition). For this reason it is not always possible to reach the convergence of the Shadow Filter and, as a consequence, to obtain an accurate estimate of the moving objects in the scene. In the future we want to cope with this limitation introducing an analysis of the structure of the scene, e.g., clustering points with similar dynamics [16] or adopting a Tracking-by-Detection approach [17], to introduce enough constrains to reduce the uncertainty associated with each point. Moreover we plan to investigate a possible extension based on the integration with an IMU to remove the constraints over the first frame, the need to perceive enough static features in each image frame, and to allow a direct estimate of the real scale.

VI. ACKNOWLEDGMENTS

This work has been partially supported by the European Commission, Sixth Framework Programme, Information Society Technologies: Contract Number FP6-045144 (RAWSEEDS), and by Italian Institute of Technology (IIT) grant.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, June 2006.

[2] D. Hahnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *IEEE ICRA*, 2003.

[3] J. Neira, A. Davison, and J. Leonard, "Guest editorial, special issue in visual slam," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 929–931, October 2008.

[4] J. Montiel, J. Civera, and A. Davison, "Unified inverse depth parametrization for monocular slam," in *RSS*, 2006 2006.

[5] L. Paz, J. D. Tardós, and J. Neira, "Divide and conquer: Ekf slam in o(n)," *IEEE Transactions on Robotics*, p. (to appear), 2008.

[6] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *ACM ISMAR*, 2007.

[7] C.-C. Wang, "Simultaneous localization, mapping and moving object tracking," Ph.D. dissertation, Robotics Institute Carnegie Mellon University, 2004.

[8] C. Bibby and I. Reid, "Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association," in *Proceedings of RSS*, 2007.

[9] A. Ess, B. Leibe, K. Schindler, and L. van Gool, "A mobile vision system for robust multi-person tracking," in *IEEE CVPR*. IEEE Press, June 2008.

[10] J. Sola, "Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach," Ph.D. dissertation, Institut National Politechnique de Toulouse, 2007.

[11] D. Marzorati, M. Matteucci, D. Migliore, and D. G. Sorrenti, "Monocular slam with inverse scaling parametrization," in *BMVC*, 2008, pp. 945–954.

[12] A. Davison, Y. G. Cid, and N. Kita, "Real-time 3D SLAM with wide-angle vision," in *Proc. IFAC Symposium on IAV*, Jul. 2004.

[13] R. Vidal, S. Soatto, and S. Sastry, "A factorization method for 3d multi-body motion estimation and segmentation," in *ACSS*, October 2002, pp. 1637–1646.

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

[15] S. Heuel, *Uncertain Projective Geometry: Statistical Reasoning for Polyhedral Object Reconstruction*. Springer, 2004.

[16] K. E. Ozden, K. Schindler, and L. van Gool, "Simultaneous segmentation and 3d reconstruction of monocular image sequences," in *ICCV*, October 2007.

[17] B. Leibe, K. Schindler, and L. J. V. Gool, "Coupled detection and trajectory estimation for multi-object tracking," in *ICCV*, 2007, pp. 1–8.