Combining Monocular and Stereo Vision in 6D-SLAM for the Localization of a Tracked Wheel Robot

Filipe Jesus and Rodrigo Ventura

Institute for Systems and Robotics

Instituto Superior Técnico

Av. Rovisco Pais, 1; Lisbon, PORTUGAL

filipejesus@ist.utl.pt, rodrigo.ventura@isr.ist.utl.pt

Abstract — Methods for the localization of tracked wheel robots in GPS-denied, unstructured environments, such as the ones encountered in Search and Rescue scenarios, meet several challenges. In most situations, one cannot rely on a planar ground, that would simplify localization and mapping. This paper addresses an online 6D-SLAM method for a tracked wheel robot, aiming at providing 6D pose estimates of the robot. While the robot pose is represented by a 3D position and a SO(3) orientation, the environment is mapped with natural landmarks in 3D space, autonomously collected using visual data from feature detectors. The observation model opportunistically employs features detected from either monocular and stereo vision. These features are represented using an inverse depth parametrization. The motion model uses odometry readings from motor encoders, together with orientation changes measured with an onboard IMU. A dimensional-bounded EKF (DBEKF) is introduced here, that keeps the dimension of the state bounded. A new landmark classifier using a Temporal Difference Learning methodology is used to identify undesired landmarks from the state. By forcing an upper bound to the number of landmarks in the EKF state, the computational complexity is reduced to up to a constant while not compromising its integrity. A real dataset from RAPOSA-NG, a tracked wheel robot developed for Search and Rescue missions, is used to experimentally validate the approach. This dataset encompasses a closed circuit, including stairs and non-planar ground segments.

Keywords: *Simultaneous localisation and mapping, Extended Kalman filter, Feature detector, Inverse depth parametrization, Landmark evaluation, Temporal difference learning*

I. INTRODUCTION

A common problem in Search and Rescue (SaR) scenarios is the inspection by remotely operated robots of building's close to collapse . Inspection operations have to be performed in a short amount of time, while collecting as much data as possible. Geo-referencing the collected data is an important feature that allow the SaR teams to better assess and plan their activities. However, these environments are usually GPS-denied, and thus alternative methods for localization are necessary.

Since neither localization nor a map are available in these scenarios, one possible approach is to use a Simultaneous Localization And Mapping (SLAM) method. SLAM is one of the most promising fields in robotics, aiming at tracking the location of a robot and map its surroundings using external sensor data. However, for SaR scenarios, the localization of a robot cannot be assumed to be limited to a 2D position. Instead, localization has to be framed as a full 6D problem (3D for position plus a SO(3) attitude).

This paper addresses this problem, termed 6D-SLAM, using an Extended Kalman Filter approach, together with visual landmarks tracked in 3D space. The sensors employed are a stereo camera pair, an Inertial Measurement Unit (IMU), and odometry readings. The IMU provides incremental attitude change readings, while odometry provides translational data along the robot body. The stereo camera pair is used to acquire both monocular and stereo features. On the one hand, stereo features provide relative 3D positioning of the feature, but stereo feature matching is prone to error, while on the other, monocular features do not require matching, but do not provide depth (requires parallax for depth to be indirectly estimated). The proposed approach employs opportunistically stereo or monocular features, depending on their availability. The usage of stereo features also solves the problem of map scale, common to most monocular SLAM techniques[1].

EKF, when applied to SLAM, proves to work reasonably well with distinct, well-matched observations and a small state for estimation. However, insertion of new data over time without removal increases EKF complexity, hindering its scalability over time. By memory-bounding the state, EKF complexity is assured to grow with time and, using proper classifiers, undesired features are automatically removed. A side effect from this removal procedure is that the map becomes visually sparse, but as long as it suffices the SLAM needs for stable predictions, one can use proper techniques to acquire visually more compelling maps.

This work was implemented on RAPOSA-NG, a tracked wheel robot for SaR missions (Figure 1). This robot has an adjustable frontal body, where both the IMU and camera are located. This robot was used to collect a dataset used to validate the approach. This dataset consists in a closed trajectory, including stairs, a ramp, and other obstacles, similar

---

[1]Assuming no *a priori* initialization of map scale.

to the ones found in the NIST reference scenario.

J.J. Leonard and H.F. Durrant-Whyte introduced *Simultaneous Localization and Mapping* (SLAM) terminology to the robotics field and the concept of *geometric beacons*: natural landmarks present in the environment that can be reliably observed, as well as described in terms of a concise geometric parametrization (referred in this paper simply as *landmarks*) [1]. Geometric beacons can be acquired with many different types of sensors, as long as the aforementioned qualities are maintained.

Nüchter *et al.* proposed a 6D-SLAM method using laser scan data [2]. The method relies on an EKF together with the Iterative Closest Point algorithm for scan matching. This method is capable of both localizing the robot and providing 3D reconstruction of the environment. However, it requires the processing of a massive amount of data provided by the laser scan data.

Civera and Davison proposed a real-time algorithm which recovers the location of a monocular camera over time using SLAM with a random walk motion model [3]. However, feature initialization requires more than one observation, so that a proper triangulation for an initial depth estimate can be done. Also, it needs to acquire landmarks with known depth for scale initialization. Thus, Civera and Davison presented an *inverse depth parametrization* that represents landmarks uncertainty with more accuracy than the standard XYZ parametrization [4]. The increase of accuracy can be justified by the higher degree of linearity of the inverse depth parametrization over XYZ parametrization. However, this representation over-parametrizes each landmark (6 instead of the 3 components of XYZ), increasing the EKF complexity even further. They also defined a landmark classifier that removes 50% of all predicted landmarks that should be visible but are not detected by any feature detector. This approach leads to the landmark classifier introduced in this paper. The usage of a random walk model assumes a well behaved motion with smooth linear and angular velocities over time, a condition that often fails for tracked wheel robots in non-planar grounds (e.g., stair climbing).

Pinies *et al.* proposed the usage of an IMU for vision SLAM with inverse depth parametrization [5]. In fact, having orientation changes measured with an IMU, the uncertainty of the camera location is reduced. However, it does not decrease the uncertainty when only linear motion is observed, which leads to the need of odometry inclusion presented in this paper. As for the map scale problem, in order to solve it, this paper extends the inverse depth parametrization usage for stereo vision as well.

This paper is organized as follows: Section II describes the state representation as well as the observation and motion model used during EKF. Section III introduces a landmark classifier that proves to be effective to measure each landmark contribution to the state. This classifier will then be applied to the EKF in order to memory-bound the state and upper limit the EKF complexity. Section IV shows experimental results in realistic environments using RAPOSA-NG, while Section V concludes the paper.
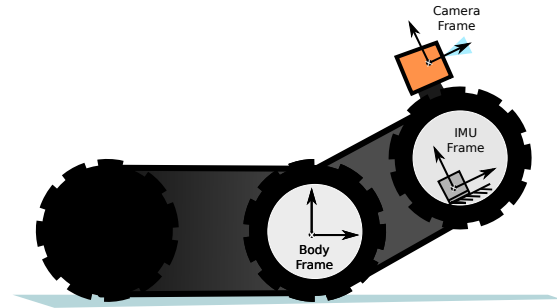

Fig. 1. Body, IMU and Camera frames are chosen depending on the robot configuration.

II. State and Model Definitions

We address the problem of simultaneously estimating the robot pose and landmark positions (SLAM) using a probabilistic approach based on the Extended Kalman Filter. The state of this filter encompasses both the robot pose and the landmark positions. The motion model is used in the predict step, while the observation model is used in the update step, as usual [1].

A. State Representation

The SLAM algorithm estimates the pose of the camera frame with respect to a world frame. The camera frame is considered to be located at the midpoint between the two stereo cameras (Figures 1 and 2).

The EKF state is defined as,

$$s_t = \begin{pmatrix} r_t^\top & q_t^\top & y_1^\top & \cdots & y_n^\top \end{pmatrix}^\top, \qquad (1)$$

where vector r_t and unit quaternion q_t represent the camera position and attitude (*i.e.,* pose of the camera frame) in the world frame at time t. All y_i from $i \in \{0, \ldots, n\}$ correspond to 3D point landmarks represented using an inverse depth parametrization,

$$y_i = \begin{pmatrix} X_i^o\ Y_i^o\ Z_i^o\ \theta_i\ \phi_i\ p_i \end{pmatrix}^\top \qquad (2)$$

where $o_i = (X_i^o, Y_i^o, Z_i^o)^\top$ is an arbitrary point in XYZ, θ_i and ϕ_i are the azimuth and elevation of the semi-ray that crosses both this point and the landmark in the world frame, and p_i is the inverse of the distance between o_i and the landmark. From this azimuth and elevation one can obtain a unit vector m in the world frame; thus landmark y_i can be expressed as

$$y_i = o_i + \frac{1}{p_i}\, m. \qquad (3)$$

This parametrization (see Figure 2) is capable of representing any landmark in space. Usually, the arbitrary point corresponds to the focal point of the camera when the landmark was first observed. While this parametrization has more degrees of freedom than necessary (6 instead of 3), it has interesting properties regarding linearity over EKF [4].

Three different frames are defined for the robot for each iteration t. Figure 1 shows the locations of these frames.

1) **Camera frame**, representing the camera pose at iteration t, as defined above;
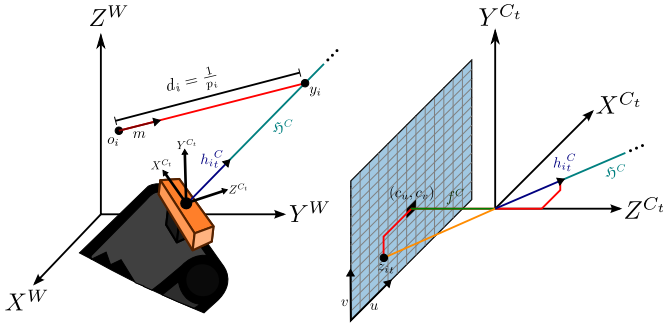
Fig. 2. Observation Model for a landmark y_i. Left: landmark y_i is parametrized by an arbitrary point o_i in the space, together with a unit vector m and a depth d_i; this unit vector is described by an azimuth and elevation angles. The observation model consists of the directional vector $h_{i_t}^C$ defined from the camera frame to the landmark. Right: the vector $h_{i_t}^C$ is then used to obtain the projection point z_{i_t} in the camera plane, which is then converted to a pixel position (u, v).

2) **Body frame**, representing the robot body pose at iteration t. In RAPOSA-NG, the transformation between this frame and the previous one depends solely on the angle of the frontal body of the robot;

3) **IMU frame,** representing the IMU pose at instant t. The angular velocity ω^{imu} can be modelled through the IMU gyroscopes. In RAPOSA-NG, this frame is attached to the frontal body.

B. Observation Model

The observation model describes how each feature is perceived by the sensors. It is used in the EKF to update the state estimate according to sensor data. Each feature can be perceived in either stereo by both cameras, or mono by only one of the cameras. In any case, the observation model provides the expected pixel position of each feature, for each one of the cameras. This computation assumes both cameras share a common image plane and is performed in two steps:

1) For each landmark y_i in state, compute a directional vector $h_{i_t}^C$ in the camera frame that points from the camera position to the landmark position (Figure 2, left);

2) Using the Pinhole Camera Model, for each landmark y_i situated in front of the camera, project the landmark position along the directional vector $h_{i_t}^C$ to the common image plane in order to get the expected pixel position (u, v) (Figure 2, right). From (u, v) and knowing the baseline of the stereo camera, compute the expected pixel position relative to each camera.

Note that this model is common to both monocular and stereo feature detection.

The pinhole model assumes a single camera with no lenses, nor aperture radius. It does not model any type of image distortion or blur present in every camera. For this paper, information retrieved for observation analysis passed through a correction process using camera's proprietary software[2] before being used by the EKF, returning an undistorted image with

---

[2]Triclops library from PointGrey.

known intrinsic parameters, while maintaining a wide visual range. This software also rectifies each pair of stereo images, by projecting them to a common image plane [6]. If no software correction is available, distortion can be compensated with proper models using distortion parameters intrinsic to the camera, retrieved through calibration methods.

An horizontal stereo camera is used in this paper to acquire image data from two different sources. Since all images are properly rectified, a given pair of features from both cameras only correspond to the same landmark if they both share the same horizontal axis. This rectification also results in a pair of images with the same size and intrinsic parameters.

C. Motion Model

The motion model employs the odometry readings to estimate linear movement along the body frame and IMU readings to estimate incremental rotations of the robot. In RAPOSA-NG the IMU is mounted on the frontal body, and thus the IMU frame sharing the same attitude as the camera frame.

From odometry the robot obtains linear movement along the body frame, by averaging the velocity of both tracks. Differential movement is discarded, since tracked wheel robots provide unreliable angular movement measurements from odometry. From the IMU, the robot obtains attitude changes. These changes are modeled as an angular velocity ω^{imu}, defined by

$$\omega_t^{imu} = \omega_t^{gyro} + \omega_t^{bias} + \omega_t^{\epsilon}, \tag{4}$$

where ω_t^{gyro} is the angular velocity retrieved from the IMU, ω_t^{bias} is the bias error normally associated with most IMUs (if the IMU uses optical or MEMS technology and is calibrated, it can be assumed no ω_t^{bias} for some period of time [7]) and ω_t^{ϵ} is a normally distributed error with zero-mean. From ω^{imu} one can obtain an incremental rotation q_t^{imu} using a zeroth-order integrator as described in [8].

The frame transformations among the frames defined in section II-A propagate these movement measurements, as well as their uncertainties (as covariances), to the camera frame.

D. Feature Initialization

Over time, visual observations are made and new landmarks are inserted into the state from observed visual features. Many criteria can be used to establish when new landmarks should be inserted and how many. For instance, one can add a new landmark every time a visual feature is observed that does not match any landmark in the state. However, doing so is computationally ineffective as it fills the state in a short time if no landmark removal procedure is performed.

Assuming the usage of the stereo camera, one can acquire monocular features either from the left or from the right camera. Also, some features acquired from both cameras correspond to the same landmark, resulting in a stereo feature. Depending on whether the new landmark in the state results from a monocular feature or from a stereo feature, two different initializations are introduced:

1) **From a monocular observation:** If a new landmark y_{n+1} is to be inserted into the state from a feature

detected by only one of the cameras, first a directional vector for the respective camera frame is computed using the Pinhole Camera Model. The $o_i = (X_i^o, Y_i^o, Z_i^o)^\top$ is set to the respective camera center, and θ_i and ϕ_i are set to the azimuth and elevation of the semi-ray that crosses this point and the feature location in image plane.

It is impossible to gain depth information from just one observation, thus an initial arbitrary value $p_{initial}$ serves as an initial estimation for the inverse depth given enough uncertainty. This parametrization is approximately linear along the corresponding semi-ray, allowing the EKF to sustain and correct large errors for the depth estimation.

2) **From a stereo observation:** Using epipolar geometry, one can compute the landmark parameters for the inverse depth parametrization, but including a measured depth, rather than a default value. The o_i is set to the camera frame origin, θ_i and ϕ_i are set to the azimuth and elevation of the semi-ray that crosses the camera frame and the landmark XYZ position extracted from stereo.

III. DIMENSIONAL-BOUNDED EKF (DBEKF)

One of the major problems regarding the Extended Kalman Filter is the fact that its computational complexity increases over a quadratic order with the number of landmarks. Since the state is represented by the camera pose and set of landmarks representing a map. The number of entries in the state vector s_t is $7 + 6\,n_l$ where n_l corresponds to the number of landmarks in the state. It has near double the number of entries when compared to all landmarks represented in XYZ coordinates. It is suggested by Civera et al [4] to convert a landmark inverse depth parametrization to XYZ in the state when the error covariance is low, but it does not interfere with the squared growth in the computational complexity of EKF.

By upper limiting the number of landmarks in the state, EKFs computational complexity becomes upper bounded. However, only upper limiting without any criterion to remove old landmarks prevents the EKF filter to acquire new features. As such, new landmarks are only added when needed and those state landmarks that are not contributing to reduce uncertainty should be removed. Since in practice, it is very difficult (although possible) to process loop closure with visual data retrieved from feature detectors when revisited in a different perspective, it is of no priority to keep old landmarks in state. However, the SLAM does not have to eliminate any landmarks unless it is needed, and a criterion for when new landmarks are needed must be defined as well.

This paper introduces a *Dimensional-Bounded Extended Kalman Filter* (DBEKF) which equips EKF with criteria for landmarks insertion and removal. Figure 3 shows a flowchart of the algorithm. Depending on the available data, the predict and/or update steps of the EKF are performed, as usual. Then, it is evaluated whether to remove and/or add new features, according to a criteria defined below. For that, a Landmark Classifier has to be introduced first.
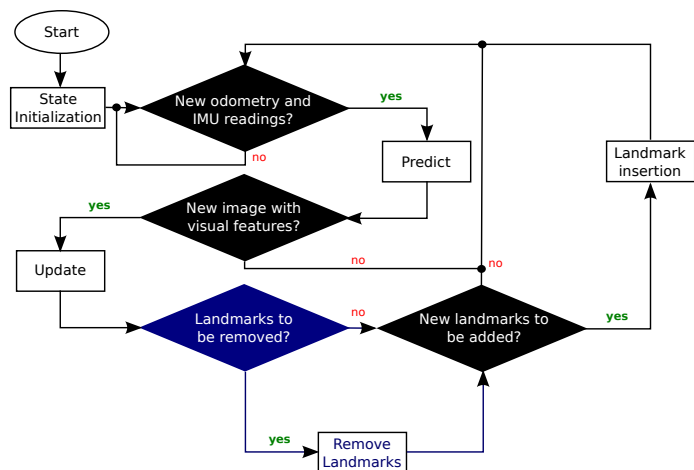

Fig. 3. Flowchart of the DBEKF algorithm.

A. Landmark Classifier

For the DBEKF, a landmark y_i is said to be *visible* in state s_t, $y_i \in \mathbf{V}_{\mathbf{s_t}}$, if it is within the field of view of the camera from state s_t. Also, y_i is *detected* at iteration t, $y_i \in \mathbf{D_t}$, if the feature detector points out a corresponding feature. In a perfect scenario without any physical occlusions, $\mathbf{V}_{\mathbf{s_t}} = \mathbf{D_t}$, that is, if the landmark is *visible* it should be *detected*. However, feature detectors are prone to error: descriptors can fail to point out some correspondences and miss features from being detected. These inaccuracies are crucial to classify each landmark's usability in the state. Since it is assumed that no landmarks have physical occlusions, a *visible* but not *detected* landmark can only represent a failed match[3]. In these cases, failed matches promote the corresponding landmark to be removed from the state.

A *Temporal Difference Learning* approach is used to predict a measure of the *utility*, u_t^i, of each landmark at iteration t:

$$u_t^i = \begin{cases} G\,u_{t-1}^i + (1 - G)\,\mathbf{1}_{\mathbf{D_t}}^i & \text{if} \quad y_i \in \mathbf{V}_{\mathbf{s_t}} \\ u_{t-1}^i & \text{otherwise.} \end{cases} \quad (5)$$

where G is a arbitrary weight set by the user and the indicator function ($G \in [0, 1]$), $\mathbf{1}_{\mathbf{D}_{\mathbf{s_t}}}^i$, is defined for detectability,

$$\mathbf{1}_{\mathbf{D}_{\mathbf{s_t}}}^i = \begin{cases} 1 & \text{if} \quad y_i \in \mathbf{D}_{\mathbf{s_t}} \\ 0 & \text{else} \end{cases}. \quad (6)$$

The lower the utility of a landmark, more likely it is to be removed from the state. The initial value for utility is $u_0^i = 1$. Due to (5) and (6), $u_t^i \in [0, 1]$.

Regarding the weight G, it represents the influence at iteration t of $\mathbf{1}_{\mathbf{D}_{\mathbf{s_t}}}^i$ over u_t^i. The lower G is, higher the influence. If $G = 0$, the utility at iteration t assumes the same value as $\mathbf{1}_{\mathbf{D}_{\mathbf{s_t}}}^i$. If $G = 1$, the utility stays equal to the initial value u_0^i, having no influence from $\mathbf{1}_{\mathbf{D}_{\mathbf{s_t}}}^i$.

---

[3]In the sense of matching between features from the camera and features stored in landmarks. Note to confuse with the stereo matching among the two cameras.

B. Landmark Removal

The landmark removal procedure is composed of three criteria:

1) **Utility Threshold:** when u_t^i reaches a value below a threshold $T_l \in [0, 1]$ at iteration t, landmark i is discarded from the state;
2) **Negative Inverse Depth:** all features with negative inverse depth (e.g., due to a feature mismatch) are automatically discarded from the state;
3) **Emergency Removal**: if the amount of matched landmarks m_l is below a threshold T_e, the N_e oldest landmarks are removed from the state, where

$$N_e = T_e - m_l. \tag{7}$$

C. New Landmark Insertion

As of new landmark insertion, the only criterion is to add n_{nl} new landmarks to the state

$$n_{nl} = min(M_l - n_l, \ n_{feat}), \tag{8}$$

where M_l is the maximum number of landmarks imposed by the user to the DBEKF, n_l is the current number of landmarks in the state and n_{feat} is the number of features without landmark correspondence given an observation at iteration t.

IV. RESULTS

All experimental results presented in this paper are from a single dataset made with RAPOSA-NG. The dataset is the result of a ROS[4] log file recording during operation. It contains odometry readings from left track, right track and inclination arm position at 15Hz each, angular velocity readings from IMU at 30Hz, rectified images from both cameras at 15Hz, and all features retrieved from image readings using feature detector ORB at 15Hz. Unless otherwise stated, all tests performed with DBEKF have an upper bound of $M_l = 80$ landmarks in the state, an utility weight factor of $G = 0.8$, an utility threshold of $T = 0.01$ and a minimal number of matched landmarks per observation of $m_l = 10$. Results slightly vary for different runs, since the log file is played in real time and new landmarks are chosen randomly.

For the experimental results, RAPOSA-NG performs a near-rectangular trip of 4.7×5.5 meters in a soccer field with obstacles, as shown in Figure 4. During the experiment, RAPOSA-NG climbs up a set of stairs with 0.62 meters of height and goes back to the floor from two ramps. Two sideway inclinations are also present, where RAPOSA-NG rolls when travelling over them.

Figure 5 presents a qualitative evaluation of the trajectories estimated by DBEKF, in comparison with using odometry alone. Three configurations are evaluated:

1) using all features as monocular features, thus being equivalent to monocular SLAM;
2) using only stereo features, where features observations are made in 3D; note however that all features not stereo matched are ignored;

[4]http://www.ros.org

Fig. 4. Scenario used for experimental results. The indicated line represents the trajectory travelled by RAPOSA-NG during the experiment. The starting point is marked with black/yellow strips, and the path proceeds clockwise (with respect to the top view).

3) using both mono and stereo features opportunistically, as proposed in this paper.

In all cases, the algorithm was capable of correcting the errors induced by odometry, up to a point, being able to return to the initial position (note also that no closure detection algorithm was employed). The high landmark covariances observed in both configurations 1 and 3 are due to the monocular feature initialization and update, that induces a high uncertainty on landmark depth. Depth uncertainty is much smaller in both configuration 2 and in the subset of stereo features in configuration 3.

A quantitative comparison between the above three configurations was made. The performance metric chosen was the trace of position estimation covariance matrix (the corresponding 3×3 submatrix of the state covariance). Recall that the trace of a square matrix equals the sum of its eigenvalues. In the case of a covariance matrix, its eigenvalues correspond to the variances along the principal axes. Since the results vary slightly over different runs over the log file, quantitative results were collected over several runs. An amount of 10 runs were used to obtain average and standard deviation values for the performance metric. The results are shown in Figure 6. These results shows that the combination of mono and stereo features in configuration 3 outperform the other two configurations. This provides empirical support to the advantages of the proposed approach.

V. CONCLUSIONS

This paper presented a localization method for a tracked wheel robot, targeting unstructured, GPS-denied environments, using vision as a primary sensor, together with odometry and IMU readings. The usage of both cameras as stereo vision decreases the uncertainty from all landmarks and allows a better initialization for the SLAM algorithm, but the lack of stereo features may offer some problems to the SLAM problem if no other type of observations are used. In this paper a method for the usage of both stereo and monocular features as observations is proposed. The more stereo features are available, the lower the localization uncertainty will be. Moreover, the presence of stereo features solves the map scale problem (referred in Section I). From the presented results, it
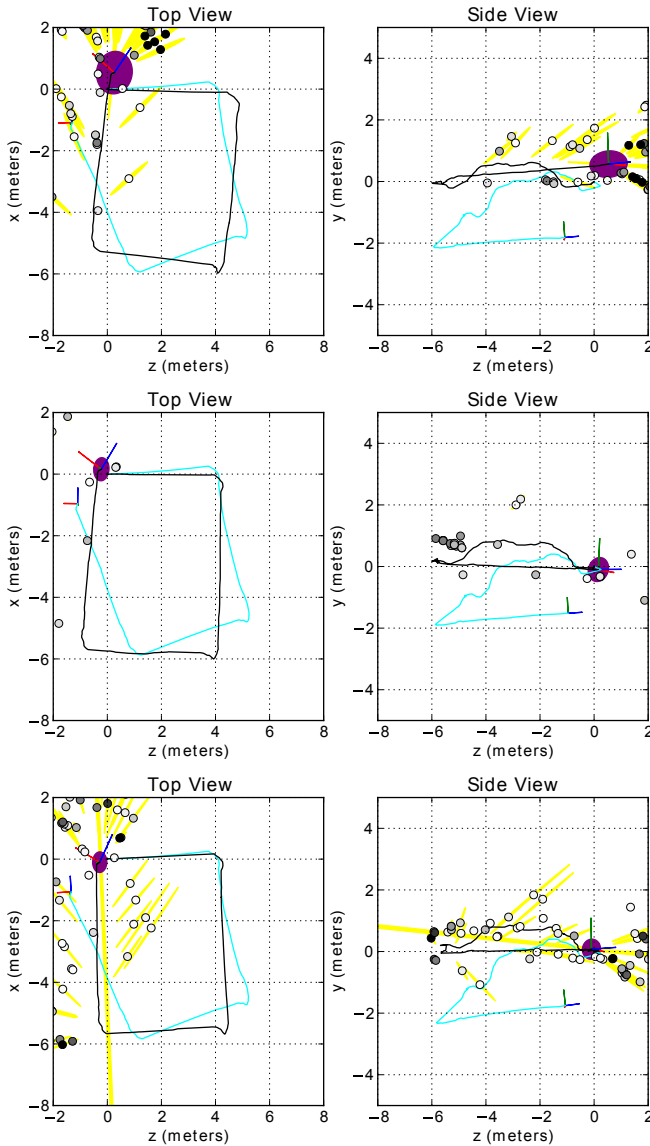
Fig. 5. Experimental results using DBEKF with mono-only (top), stereo-only (middle) and stereo plus mono solution (bottom). These correspond to configurations 1, 2, and 3 as described in the main text. The trajectories shown are: in cyan/grey is the camera trajectory with only odometry and IMU, while the black one used SLAM estimation. Covariance for the final position is also shown (in violet), as well as the final covariances of the landmarks in the state (in yellow).

is clear that using both monocular and stereo observations in the way introduced here increases the overall quality of SLAM over monocular only or stereo only observations. Experimental results have shown that the proposed method outperforms the exclusive usage of either monocular or stereo features.

Although the EKF has been extensively used to solve the SLAM problem, its computational complexity grows unbounded with the number of landmarks. This paper showed that, with DBEKF, one can achieve good estimations with constant complexity when removing landmarks from state according to a utility evaluation criterion.

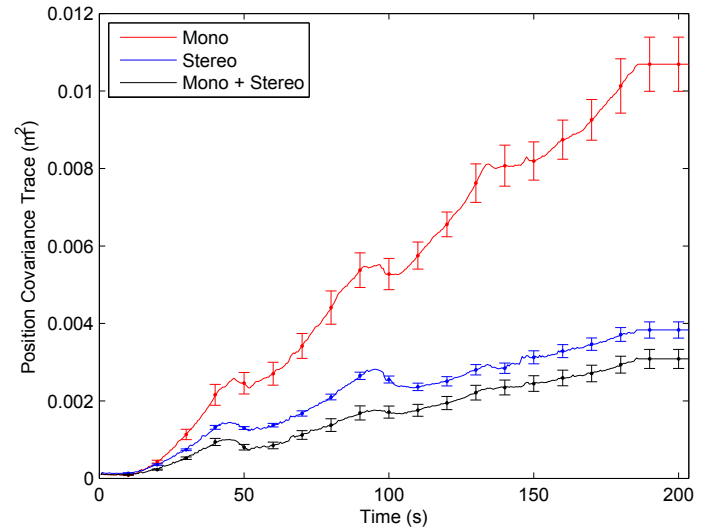Some open challenges that can be addressed in future work

Fig. 6. Evolution of the performance metric (position covariance, lower is better) using DBEKF along time. The different traces correspond to the use of mono features only, of stereo features only, and of all features. The values are averages of the metric over 10 runs, together with standard deviances as error bars.

include the usage of sub-mapping techniques to store old features, instead of removing them from the state. This allows keeping all features encountered so far, without compromising the DBEKF performance. Older features could also be used to explicitly address the loop closure problem.

REFERENCES

[1] J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, 1991, pp. 1442–1447.

[2] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM — 3D mapping outdoor environments," *Journal of Field Robotics*, vol. 24, no. 8–9, pp. 699–722, 2007.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera slam," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, June 2007.

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

[5] P. Pinies, T. Lupton, S. Sukkarieh, and J. Tardos, "Inertial aiding of inverse depth SLAM using a monocular camera," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 2797–2802.

[6] D. Oram, "Rectification for any epipolar geometry," in *BMVC'01*, 2001, pp. 653–662.

[7] O. J. Woodman, "An introduction to inertial navigation," University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-696, Aug. 2007.

[8] N. Trawny and S. Roumeliotis, "Indirect kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep., 2005.