Simultaneous Localization and Mapping using Stereo and Mono Vision for Tracked Wheel Robots*

Filipe Jesus\(^1\) and Rodrigo Ventura\(^2\)

Abstract—This paper addresses an online 6D SLAM for a tracked wheel robot in an unknown and unstructured environment. While the robot pose is represented by a 3D position and SO(3) orientation, the environment is mapped with natural landmarks in 3D space, autonomously collected using visual data from feature detectors. A motion model using odometry readings from motor encoders and orientation changes measured with an IMU is introduced, as well as an observation model with a novel approach that copes with both mono and stereo observations.

A new landmark classifier using a Temporal Difference Learning methodology is introduced. This classifier is applied for landmark removal, identifying all undesired landmarks for estimation. The main novelty introduced is a Dimensional-bounded EKF that, with the aforementioned classifier and proper criteria for landmark removal and insertion, forces an upper bound to the number of landmarks in the EKF state, reducing the computational complexity up to a constant while not compromising its integrity.

All experimental work was done using real data from RAPOSA-NG, a tracked wheel robot developed for Search and Rescue missions.

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

I. INTRODUCTION

SLAM is one of the most promising fields in robotics, aiming to track the location of a robot and map its surroundings using external sensor data. EKF, when applied to SLAM, proves to work reasonably well with distinct, well-matched observations and a small state for estimation. However, blind insertion of new data over time without removal increases EKF complexity, making it obsolete for large amount of data estimation.

By memory-bounding the state, EKF complexity is assured to stay bellow a threshold 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. Since most SaR robots perform motion within irregular terrains, this paper focuses on estimating both position in a 3D euclidean space and a SO(3) orientation. It uses an IMU to measure orientation changes and encodes odometry from both wheels to measure translation changes over a set of states. Also, this paper introduces an elegant way to introduce landmarks in the state from both monocular and stereo visualizations using the inverse depth parametrization. All landmarks are treated the same way, regardless of their origin. While monocular observations have no depth information, they allow the estimation of depth through parallax changes during time.

This paper is organized as follows: Section II reviews the literature. Section III describes the state representation as well as the observation and motion model used during EKF. Section IV 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, as explained in section V. Section VI shows test results in realistic environments using RAPOSA-NG. Section VIII concludes the paper.

II. LITERATURE REVIEW

It was during the 80s that R.C. Smith and P. Cheeseman presented a stochastic way of measuring a number of spatial relationships with a common world frame by compounding successive states modelled with uncertainty [1]. Extended Kalman Filter (EKF) was then applied to estimate it in an iterative way, computing a proper gain that would weight both observable and prior spatial relationships. Their remarkable work gave rise to a probabilistic approach for the robotics field and many researchers strived to apply it to a real case scenario. J.J. Leonard and H.F. Durrant-Whyte not only accomplished that in 1991 with a real robot in a known environment, but also introduced Simultaneous Localization and Mapping (SLAM) terminology to the robotics field and the concept of geometric beacons: natural landmarks present in the environment that could be successively observed with reliability and be well described in terms of a concise geometric parametrization (referred in this paper simply as landmarks) [2]. Geometric beacons can be acquired with many different types of sensors, as long as the aforementioned qualities are maintained.

Civera and Davison proposed a real-time algorithm which recovers the location of a monocular camera over time using SLAM with a random walk model [3]. However, feature initialization requires more than one observation and a proper triangulation for an initial depth estimate. Also, it needs

\(^*\)This work was supported by project FCT [PEst-OE/EEI/LA0009/2011]

\(^1\)Filipe Jesus is with Institute for Systems and Robotics, Instituto Superior Técnico, TULisbon, Portugal filipe.moj at gmail.com

\(^2\)R. Ventura is with Institute for Systems and Robotics, Instituto Superior Técnico, TULisbon, Portugal rodrigo.ventura at isr.ist.utl.pt
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 parametrizes each landmark with double the parameters, 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 per iteration. This approach lead to the landmark classifier introduced in this thesis. The usage of a random walk model assumes a well behaved motion with smooth linear and angular velocities for all time, a condition that surely fails when, for instance, a robot climbs up a set of stairs.

Pinhies, Lupion, et al. added the usage of an IMU to the vision SLAM with inverse depth parametrization [5]. In fact, having orientation changes measured with an IMU, the uncertainty of the camera location is better modelled. 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 scalability problem, in order to solve it, this paper extends the inverse depth parametrization usage for stereo vision as well.

III. STATE REPRESENTATION

A. STATE REPRESENTATION

The state is defined as,

\[ s_t = \begin{pmatrix} x_t^T & m^T \end{pmatrix}^T, \]

that is, both camera pose, \( x_t \), and the map in the form of a sequence of landmark positions, \( m \), with respect to the world frame are present for estimation. It is assumed that the environment surrounding the robot is static, making \( m \) invariable with time.

The camera pose state at time \( t \), \( x_t \), provides the position and orientation of the camera within the map,

\[ x_t = \begin{pmatrix} r_t^T & q_t^T \end{pmatrix}^T \]

\[ r_t = \begin{pmatrix} x_{1t} & y_{1t} & z_{1t} \end{pmatrix}^T, \quad q_t = \begin{pmatrix} q_{1t} & q_{2t} & q_{3t} & q_{4t} \end{pmatrix}^T. \]

The vector \( r_t \) and the quaternion \( q_t \) describe both camera position in the world frame and spatial rotation from the world frame to the camera frame at time \( t \), respectively. Although less intuitive, quaternion representation for orientation suffers no singularity, avoiding the gimbal lock problem from Euler Angles. Also, they are advantageous over rotation matrices for real time implementations, as quaternions are more compact, numerically stable and computationally efficient.

This SLAM maps the environment by observing point features from a visual camera and introducing them on the state as landmarks,

\[ m = \begin{pmatrix} y_1^T & \ldots & y_n^T \end{pmatrix}^T. \]

Fig. 1: Observation Model for landmark \( y_i \).

Each landmark \( i \) is represented using inverse depth parametrization,

\[ y_i = \begin{pmatrix} o_i \\ \theta_i \\ \phi_i \\ p_i \end{pmatrix} = \begin{pmatrix} \arctan(h_{z,1}, h_{x,1}) \\ \arctan(-h_{z,1}, \sqrt{h_{x,1}^2 + h_{z,1}^2}) \\ \sqrt{h_{x,1}^2 + h_{z,1}^2} \\ p_i \end{pmatrix}, \]

\[ h_i = P_i - r \]

where \( P_i \) is the landmark pose in XYZ, \( o_i \) is the camera pose from when the landmark was first seen, \( \theta_i \) and \( \phi_i \) are the azimuth and elevation of the semi-ray that crosses both \( o_i \) and the landmark in the world frame, and \( p_i \) is the inverse of the distance between \( o_i \) and \( P_i \).

B. OBSERVATION MODEL

The Observation Model adopted for this approach computes \( z_t \), a vector with all \( n \) features correspondent to each landmark in state, given the camera pose at instant \( t \),

\[ z_t = \begin{pmatrix} z_{1t}^T & \ldots & z_{nt}^T \end{pmatrix}^T. \]

Assuming that the camera focal point is located over the camera frame origin and shares the same orientation, this model is composed of two steps:

1) For each landmark \( y_i \) in state, compute a directional vector in the camera frame that points from the camera position to the landmark position. All landmarks within the interval \( \mathcal{S}^C \) result in the same semi-ray for the same camera pose, and therefore, share the same visual observation. The interval \( \mathcal{S}^C \) in the camera frame can be parametrized as

\[ \mathcal{S}^C = \{ R_{q_i^*} j (o_i - r_t + 1 \frac{1}{p_i} m_i) \mid j \in [0, +\infty] \} \]

where \( R_{q_i^*} \) is the rotation matrix that rotates the world frame to the camera frame (from \( q_i^* \), conjugate, \( q_i^* \)). A possible directional vector that can be easily computed is \( h_{1t}^C \), defined as,

\[ h_{1t}^C = R_{q_i^*} h_{1t}, \quad \text{where} \]

\[ h_{1t} = (p_i (o_i - r_t) + m_i). \]
Choosing $h_{i_t}^{C}$ from $\mathcal{S}^C$ allows the observation model to measure landmarks with infinite depth, since

$$p_i = 0 : \quad h_{i_t}^{C} = R_{q_i} m_i,$$  \hfill (11)

2) Using the Pinhole Camera Model, for each landmark $i$ situated in front of the camera, project $h_{i_t}^{C}$ on the camera’s image plane that is orthogonal to the camera’s optical axis. Knowing the focal length, $f$, the scale factors relating pixel coordinates to meters in the image plane, $m_u$ and $m_v$, and the location of the principal point in pixels, $c_u$ and $c_v$, the directional vector $h_{i_t}^{C}$ can be directly mapped onto pixel coordinates, $z_{i_t}$, by

$$\begin{pmatrix} z_{u,i_t} \\ z_{v,i_t} \end{pmatrix} = \begin{pmatrix} c_u \\ c_v \end{pmatrix} - \frac{f}{h_{z,i_t}} \begin{pmatrix} m_u h_{x,i_t} \\ m_v h_{y,i_t} \end{pmatrix} + \delta_{i_t} \tag{12}$$

where

$$\delta_{i_t} \sim \mathcal{N}(0, \sigma^2_{\text{pixel}} I_{(2 \times 2)}).$$  \hfill (13)

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 before being used by the EKF, returning an undistorted image with 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 [6].

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.

A set of coordinate frames, $(X^{CL_t}, Y^{CL_t}, Z^{CL_t})$ and $(X^{CR_t}, Y^{CR_t}, Z^{CR_t})$, are defined for the left and right camera, respectively. After the rectification process, both right and left camera frames share the same orientation as the camera frame and are displaced by $b$ along the $X^{C_t}$ axis.

To simplify the formalism, a parameter $k^{LR}$ is introduced, where

$$k^{LR} = \begin{cases} 0, & \text{from left (L) camera} \\ 1, & \text{from right (R) camera} \end{cases}$$  \hfill (14)

Both directional vectors from left and right cameras, $h_{i_t}^{CL}$ and $h_{i_t}^{CR}$, can easily computed from $h_{i_t}^{C}$,

$$h_{i_t}^{CL/CR} = h_{i_t}^{C} - (-1)^{k^{LR}} h_{i_t}^{C}$$  \hfill (15)

where

$$h_{i_t}^{C} = p_i \bar{b} \quad \text{and} \quad \bar{b} = (b \ 0 \ 0)^T.$$  \hfill (16)

With the pinhole model, one can either model an observation from both cameras,

$$z_{\text{Stereo}}^{i_t} = \begin{pmatrix} z_{u,i_t}^{L} \\ z_{v,i_t}^{L} \\ z_{u,i_t}^{R} \\ z_{v,i_t}^{R} \end{pmatrix} = \begin{pmatrix} f C m_u p_i \bar{b} \\ -b \\ 0 \\ 0 \end{pmatrix},$$  \hfill (17)

or from left or right camera only,

$$z_{\text{Left}}^{i_t} = \begin{pmatrix} z_{u,i_t}^{L} \\ z_{v,i_t}^{L} \end{pmatrix}, \quad \quad z_{\text{Right}}^{i_t} = \begin{pmatrix} z_{u,i_t}^{R} \\ z_{v,i_t}^{R} \end{pmatrix}. \tag{18}$$

C. MOTION MODEL

For the Motion Model, it is assumed that the SaR robot moves in a 3D space and uses an IMU and odometry to help predict its location over time.

Three different frames are defined for the robot for each iteration $t$. Figure 3 shows a possible distribution for all three frames.

1) **Camera frame.** $(X^{C_t}, Y^{C_t}, Z^{C_t})$: it represents the camera pose at iteration $t$ and the frame this SLAM is locating. The camera pose related to the world frame is the camera state $x_t$. The corresponding homogeneous transform is

$$T_W^{C_t} = \begin{bmatrix} R_{q_t} & r_t \\ 0 & 1 \end{bmatrix},$$  \hfill (19)

where $R_{q_t}$ is the rotation orthogonal matrix acquired from the quaternion $q_t$ in state.
2) **Body frame.** \((X^B_t, Y^B_t, Z^B_t)\): it represents the robot body pose at iteration \(t\). The spatial translation from \(B_t\) to \(B_{t+1}\) in the world frame, \(r_{B_{t+1}}^B\), can be obtained using a simple odometry motion model and by integrating rotary encoder information from each wheel \((r_{e}^w, r_e^l)\). Knowing the time difference between iterations \(t\) and \(t+1\), \(\tau_{t,t+1}\),

\[
r_{e}^r = r_{e}^{r,t} - r_{e}^{r,t} + \tau_{t,t+1} r_{e}^{r,t+1}
\]

\[
r_{e}^l = r_{e}^{l,t} - r_{e}^{l,t} + \tau_{t,t+1} r_{e}^{l,t+1},
\]

where

\[
r_{e}^{r,t+1} \sim N(0, \sigma^2_{r,odoright})
\]

\[
r_{e}^{l,t+1} \sim N(0, \sigma^2_{r,oleft}).
\]

For instance, assuming a two-wheel robot with movement \(r_{B_{t+1}}^B\) coplanar to the plane \((XOY)^B\),

\[
|r_{B_{t+1}}^B| = \frac{r_{e}^{r,t} + r_{e}^{l,t}}{2}
\]

\[
r_{B_{t+1}}^B = \left| r_{B_{t+1}}^B \right| \cos(\frac{r_{e}^{r,t} - r_{e}^{l,t}}{2})
\]

\[
r_{B_{t+1}}^B \sin(\frac{r_{e}^{r,t} - r_{e}^{l,t}}{2}) \frac{L_{odo}}{0}
\]

where \(|r_{B_{t+1}}^B|\) is the L2-norm of \(|r_{B_{t+1}}^B|\) and \(L_{odo}\) is the size of the line segment that crosses both wheels and the robot frame origin. Unfortunately, a simple two-wheeled motion model for movement will not be accurate enough for SaR robots like RAPOSA-NG that use tracked wheels to grant more kinetic friction with inclined surfaces and stairs, varying \(L_{odo}\) with unmeasured external factors over time. As such, since this model fuses information from an IMU, it will not consider the resulting body frame orientation from encoders information.

3) **IMU frame.** \((X^I_t, Y^I_t, Z^I_t)\): it represents the IMU pose at instant \(t\). The angular velocity \(w_{imu}^I\) can be modelled through the IMU gyroscopes,

\[
w_{imu}^I = w_{i}^{qyro} + w_{i}^{bias} + w_{i}^{c}
\]

where \(w_{i}^{qyro}\) is the angular velocity retrieved from the IMU, \(w_{i}^{bias}\) 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 \(w_{i}^{bias}\) for some period of time [7]) and \(w_{i}^{c}\) is a normally distributed error with zero-mean,

\[
w_{i}^{c} \sim N(0, \sigma^2_{w,i}^{imu}).
\]

Knowing the time difference between instants \(t\) and \(t+1\), \(\tau_{t,t+1}\), and assuming constant velocity \(w_{imu}^I\) one can compute the spatial rotation from frame \(I_t\) to \(I_{t+1}\) using quaternion notation \(q_{imu}^I\),

\[
q_{imu}^I = \cos(2 |w_{imu}^I| \tau_{t,t+1})
\]

\[
\frac{q_{imu}^I \ q_{imu}^I \ q_{imu}^I}{w_{imu}^I \ w_{imu}^I \ w_{imu}^I} = \sin(2 |w_{imu}^I| \tau_{t,t+1})
\]

This model assumes an IMU installed at any part of the robot, given that all transformations from the robot camera frame or body are well known at each instant. It can be retrieved through servo feedback and proper encoders. Using homogeneous transformation matrices, the state transition can be computed as

\[
T_{C_{t+1}}^{W} = T_{C_{t}}^{W} \cdot T_{I_{t+1}}^{C_{t}} \cdot T_{I_{t+1}}^{C_{t+1}} \cdot (T_{I_{t+1}}^{C_{t}})^{-1},
\]

where both \(T_{I_{t+1}}^{C_{t}}\) and \(T_{I_{t+1}}^{C_{t+1}}\) are well known transforms from the camera frame to the IMU frame in their respective instants. Regarding \(T_{I_{t+1}}^{C_{t}}\),

\[
T_{I_{t+1}}^{C_{t}} = T_{B_{t+1}}^{I_{t+1}} \cdot T_{B_{t+1}}^{B_{t+1}} \cdot (T_{B_{t+1+1}}^{B_{t+1}})^{-1},
\]

while the rotation \(R_{B_{t+1}}^{I_{t+1}}\) is available by converting \(q_{imu}^I\) to an orthogonal rotation matrix, the translation \(r_{I_{t+1}}^{B_{t+1}}\) is given by

\[
r_{I_{t+1}}^{B_{t+1}} = r_{I_{t+1}}^{I_{t+1}} \cdot R_{B_{t+1}}^{I_{t+1}} \cdot R_{B_{t+1}}^{B_{t+1}} \cdot R_{B_{t+1+1}}^{B_{t+1}}
\]

\[
T_{I_{t+1+1}}^{B_{t+1+1}}\]

where both \(T_{B_{t+1}}^{I_{t+1}}\) and \(T_{B_{t+1+1}}^{B_{t+1}}\) are also well known transforms from the IMU frame to the body frame in their respective instants.

**D. FEATURE INITIALIZATION**

Over time, visual observations are made and new landmarks are attached to 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 correspond to any landmark in state, but it proves to be 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 depicted in figure 2, 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 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 \(y_n\) is to be attached to the state with \(n - 1\) landmarks from a feature \(z_{n}^{L/CR}\) or \(z_{n}^{R/CR}\) alone, first a directional vector for the respective camera frame is computed using the Pinhole Camera Model from equation (12).

\[
h_{n}^{CL/CR} = \frac{L/CR}{C/CR} (C_{m} - z_{n}^{L/CR})
\]

Having \(R_{t}\) from \(q_{t}\) in state and knowing that both left and right camera frames share the same orientation
from the robot state, the directional vector can be related to the world frame by
\[ h_{nt} = R_{nt} h^{CL/CR}_{nt} \] (34)
and
\[
\begin{pmatrix}
\alpha_n \\ \theta_n \\ \phi_n \\ p_n
\end{pmatrix} =
\begin{pmatrix}
\frac{r_t + (-1)^{t\, L/R} R_{nt} b}{\arctan(h_{xn_t}, h_{yn_t})} \\
\frac{\arctan(-h_{yn_t}, h_{xn_t})}{p_{\text{initial}}}
\end{pmatrix} (35)
\]
where
\[ |h_{x\, nt}| = \sqrt{(h_{x\, nt})^2 + (h_{x\, nt})^2} \] (36)

It is impossible to gain depth information from just one observation, thus an initial arbitrary value \( p_{\text{initial}} \) 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 big errors for the depth estimation.

2) From a stereo observation: If a new landmark \( y_n \) is to be attached to the state with \( n-1 \) landmarks from a stereo pair of features, \( z_{nt}^{\text{Stereo}} \), first a directional vector from the camera frame is computed using the Pinhole Camera Model from equation (12). Since both left and right camera frames share the same distance from the camera frame is computed using the Pinhole Camera Model from equation (12). Since both left and right camera frames share the same distance from the camera frame but in opposite directions, the directional vector can be calculated as
\[ h^C_n = \left( \begin{array}{c}
\frac{1}{f_c m_u} (c_u - \frac{1}{2} \left( v_{wnt} + v_{wnl} \right)) \\
\frac{1}{f_c m_v} (c_v - z_{wnu})
\end{array} \right) \] (37)

In the same fashion as equation (34),
\[ h_{nt} = R_{nt} h^C_{nt} \] (38)
and
\[
\begin{pmatrix}
\alpha_n \\ \theta_n \\ \phi_n \\ p_n
\end{pmatrix} =
\begin{pmatrix}
\frac{r_t}{\arctan(h_{xn_t}, h_{yn_t})} \\
\frac{\arctan(-h_{yn_t}, h_{xn_t})}{p_{\text{epipolar}}}
\end{pmatrix} (39)
\]
where \( p_{\text{epipolar}} \) can be computed using epipolar geometry [8],
\[ p_{\text{epipolar}} = \frac{z_{wnl} - z_{wnu}}{2 b f_c m_u |h^C_{nt}|} \] (40)

For a state with \( n-1 \) landmarks, \( s_t \) estimation (as a normal distribution) can be updated by adding a new landmark \( y_n \) initial estimate [4],
\[ \mu_t^{\text{new}} = \left( \begin{array}{c}
\mu_t \\
y_n
\end{array} \right) \] (41)
This paper introduces a Dimensional-bounded Extended Kalman Filter (DBEKF) which encompasses EKF with special criterion for landmarks insertion and removal. For that, a new Landmark Classifier is defined.

A. Landmark Classifier

For the DBEKF, a landmark \( y_i \) is said to be visible in state \( s_t, y_i \in V_{s_t} \), if it is observable from state \( s_t \). Also, \( y_i \) is detected at iteration \( t, y_i \in D_t \), if the feature detector points out a corresponding feature. In a perfect scenario, assuming that no landmarks have physical occlusions

\[
V_{s_t} \subseteq D_t, \quad (45)
\]

that is, if the landmark is visible it should be observable. However, feature detectors are far from having a perfect behaviour: descriptors can fail to point out some correspondences and miss features from being detected. This inaccuracies are crucial to classify each landmark’s usability in state. Since it is assumed that no landmarks have physical occlusions, a visible but not observed landmark can only represent a failed match.

Since the state in EKF has only gaussian distributions representing landmark pose estimates, a Temporal Difference Learning approach [9] is used to predict a measure of the utility, \( \text{util}^t_i \), at iteration \( t \):

\[
\text{util}^t_i = \begin{cases} 
G \text{util}^{t-1}_i + (1 - G) 1^t_{D_{s_t}} & \text{if } y_i \in V_{s_t} \\
\text{util}^{t-1}_i & \text{else}
\end{cases} \quad (46)
\]

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

\[
1^t_{D_{s_t}} = \begin{cases} 
1 & \text{if } y_i \in D_{s_t} \\
0 & \text{else}
\end{cases} \quad (47)
\]

The initial value for utility is

\[
\text{util}^0_i = 1. \quad (48)
\]

From equations (46) and using the condition from (48), it can be shown that

\[
\forall_G \in [0, 1] \quad \forall_t, \quad \text{util}^t_i \in [0, 1] \quad (49)
\]

Regarding the weight \( G \), it represents the influence at iteration \( t \) of \( 1^t_{D_{s_t}} \) over \( \text{util}^t_i \). The lower \( G \) is, higher the influence. If \( G = 0 \), the utility at iteration \( t \) assumes the same value as \( 1^t_{D_{s_t}} \). If \( G = 1 \), the utility stays equal to the initial value \( \text{util}^0_i \), having no influence from \( 1^t_{D_{s_t}} \).

\[
\text{if } G = 1 : \quad \forall_t \quad \text{util}^t_i = 1 \quad (50)
\]

\[
\text{if } G = 0 : \quad \forall_t \quad \text{util}^t_i = \{0, 1\} \quad (51)
\]

\[
\text{if } G \in ]0, 1[ : \quad \forall_t \quad \text{util}^t_i = [0, 1] \quad (52)
\]

B. Landmark Removal

Assuming \( Max_l \) as the maximum number of landmarks imposed by the user to the DBEKF, the Landmark removal procedure is composed of three criteria:

1) **Utility Threshold:** Regarding the utility \( \text{util}^t_i \), one can interpret \( \text{util}^t_i = 1 \) as a maximum score and \( \text{util}^t_i = 0 \) as a minimal score (although it can only be equal to zero if \( G = 0 \)). A simple but effective approach for a landmark removal criterion is that when \( \text{util}^t_i \) reaches a value bellow \( T \) at iteration \( t \), it is discarded from the state, where \( T \in [0, 1] \).

2) **Negative Inverse Depth:** From the first-order linearisation nature of EKF, it may happen that a landmark estimation in state gets a negative inverse depth. This situation, although not common, can damage the whole process. As such, all features with negative inverse depth are automatically discarded from the state.

3) **Emergency Removal:** Although the aforementioned two criteria are enough for a landmark quality control, the risk of filling the state and not being able to add new, important, landmarks for the observation of new areas maintains. As such, a used defined value \( m_{obs} \) is presented, stating the minimal number of matched landmarks per observation. Note that

\[
m_{obs} \leq M_l \quad (53)
\]

If only \( n \) landmarks were matched and \( n < m_{obs} \), the number of landmarks to eliminate should be, at least, \( m_{obs} - n \).

Algorithm 1 shows in pseudocode how the Landmark Removal is processed.

C. New Landmark Insertion

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

\[
n_{nl} = \min(M_l - n_t, n_{feat}), \quad (54)
\]

where \( n_t \) is the current number of landmarks in state and \( n_{feat} \) is the number of features without landmark correspondence given an observation at iteration \( t \). Algorithm
Algorithm 1 Landmark Removal

for each landmark in state do
  \( k \leftarrow 0 \)
  if landmark is visible then
    if landmark was detected then
      \( \text{util}_i \leftarrow 1 + G \times (\text{util}_{i-1} - 1) \)
      \( k \leftarrow k + 1 \)
    else
      \( \text{util}_i \leftarrow G \times \text{util}_{i-1} \)
    end if
  end if
end for

\( n \leftarrow m_{\text{observations}} - k \)
for each landmark in state do
  if \( n > 0 \) then
    landmark is removed
    \( n \leftarrow n - 1 \)
  end if
end for

for each landmark in state, sorted in ascending order of \( \text{util} \) do
  if \( \text{util}_i \leq T \) or \( p_i \leq 0 \) then
    landmark is removed
  end if
end for

Algorithm 2 New Landmark Insertion

\( n \leftarrow M_l - n_l \)
for each feature in observation, sorted in descending order of strength do
  if \( n > 0 \) and did not match any landmark in state then
    state \( \leftarrow \) newlandmark from feature
    \( n \leftarrow n - 1 \)
  end if
end for

V. TESTS AND RESULTS

All test results presented in this thesis are from two different datasets made with RAPOSA-NG, both recorded in the Laboratorio de Robtica Mvel from Instituto Superior Tecnico (IST) at Campus Alameda. Both were recorded using ROSBAG application from middleware ROS \(^1\). Also, the stereo camera is attached to a Pan&Tilt structure in RAPOSA-NG that is not being used for this thesis work but makes the camera sway a little vertically when moving. While slightly observed with the camera, these small camera movements are not caught by the IMU and may incur against the condition presented by the motion model that requires both camera and IMU frames to share the same orientation for all time.

Each dataset contains odometry readings from left track, right track and inclination arm position at 10Hz 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 and descriptor ORB at 15Hz. Results may slightly vary for the same dataset in different runs. Unless it is said otherwise, all tests performed with DBEKF have an upper bound of \( M_{\text{landmarks}} = 60 \) landmarks in 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 \( M_{\text{landmarks}} = 10 \).

The datasets are as follows, both depicted in figure 6:

- Dataset “A” - Square Trip
  On this dataset, RAPOSA-NG performs a near-squared trip of \( 3 \times 3 \) meters in a soccer field full of newspaper pages, wooden planks and other sort of objects, simulating debris. Newspapers were scattered around the floor to present a larger spectrum of landmark candidates from feature detection, while messing with odometry readings when turning. The inclination arm angle is zero for all time, and the stereo camera is always parallel to the floor. The robot never stops moving, even when rotating every 90 degrees. Also, it finishes the trajectory facing the same plane from the starting point. This dataset helps to evaluate how the SLAM deals with planar trajectories when performing 3D pose estimation, as well as how the SLAM behaves when, after turning, faces an entirely new plan that requires

\(^1\)http://www.ros.org
new observations. It has a duration of 2 minutes and 48 seconds.

- **Dataset “B” - Stairs Trip**
  On this dataset, RAPOSA-NG climbs up and down a set of stairs, placed in the same scenario as dataset “A”. The stairs set has 0.615 meters of height. The distance in $Z^W$ from the starting point to the farthest point the robot reaches is of approximately 2.5 meters. Upon reaching the stairs top, RAPOSA-NG drives backwards along the same path until it reaches the starting point again. Stair climbing offers a number of challenges for SLAM: it highly affects odometry readings, it offers a huge amount of feature occlusions while climbing and the robot movement has no smoothness. This dataset has a duration of 1 minute and 38 seconds.

A. Computational load and feature removal with DBEKF

By upper limiting the number of landmarks in state by a value $M_{\text{landmarks}}$, EKF computational complexity becomes upper bounded as well. If there are enough observations to grant $M_{\text{landmarks}}$ in state for estimation at every iteration, the computational load should be constant for all time. This situation happens to all experiments presented on this thesis.

Figure 7 shows the time duration and number of removed features per DBEKF iteration with both monocular and stereo observations from datasets “A” and “B”, respectively. As expected, the computational complexity is near constant for all time during both experiments, presenting some peaks and fluctuations due to new landmark initialization, the number of feature observations per update and other processing tasks unrelated to this software. Regarding dataset “A”, one can notice some time intervals where a larger number of features are removed. These intervals happen when dataset “A”, one can notice some time intervals where a larger number of features are removed. These intervals happen when the robot finishes rotating 90 degrees and faces a new plane of observations, requiring space for new landmarks in state. It then discards older landmarks in order to acquire new ones. Dataset “B” presents some peaks regarding the number of feature observations as well, where the robot experienced drastic observation changes due to the rough movement of RAPOSA when finishing climbing up or starting to climb down the stairs.

From the average time reading in both experiments, it is clear that the SLAM algorithm fully performs in real time. However, it does not take into account the time needed for feature acquisition using feature detectors such as SURF or ORB.

Figure 8 shows the average time per iteration for different upper bounds in DBEKF for both datasets “A” and “B”. It is important that, although the computational power becomes constant, a reasonable upper bound is chosen to avoid large time intervals for the EKF that can otherwise reveal linearity problems.

B. Mono, Stereo and Hybrid SLAM using DBEKF

Both monocular and stereo observations can be used to correct the a priori state estimation in an elegant way using
Fig. 9: Pair of stereo image with visual features. Monocular features are represented with a red dot, while stereo features are represented with a green dot and matched with a green line.

Fig. 10: Number of stereo and mono features acquired with ORB for each dataset "A" and "B".

Fig. 11: SLAM results using DBEKF with dataset “A”, using only monocular observations from left camera (a), only stereo observations (b) and both monocular and stereo observations (c). Two different camera trajectories are presented in each graph: in blue is the camera trajectory with only odometry and IMU, while the black one used SLAM estimation. Both final positions have their orientation represented using RGB colors for each XYZ orientation axes, respectively. All landmarks are represented as small circumferences, and their color intensity is proportional to their utility from DBEKF (black=1, white=0). Finally, the areas in purple and yellow represent the final pose covariance and landmarks covariance, respectively.

The novelistic approach presented in this thesis. This section compares this solution over monocular only and stereo only solutions. Note that for the monocular only solution, the left camera was chosen arbitrarily for feature acquisition. Using epipolar geometry and feature descriptors, one can identify a stereo feature by matching a pair of monocular features that share the same horizontal axis.

Figure 10 shows the number of stereo and monocular features acquired with the stereo camera Bumblebee2 using ORB for both datasets “A” and “B”. While for dataset “A” the stereo acquisition seems constant during all experiment, a huge increase of stereo features (and decrease of monocular features) can be observed during the middle of dataset “B” experiment, while climbing up and down the stairs. During this period, the robot observes the upper part of the scenario, contributing with observations that suffer no perspective problems, unlike the stairs that are in close proximity with the robot. This results suggest the need of coupling both monocular only and stereo observations for state estimation.

Figures 11 and 12 present SLAM graphical results for datasets “A” and “B” using monocular only, stereo only and both monocular and stereo observations with DBEKF. Figure 13 shows the pose covariance trace through time for all the experiences aforementioned. Regarding monocular observations only, both datasets suffer from scalability problems when estimating the trajectory. While it strives to correct the pose and orientation using pallarax changes only, the pose covariance following the robot trajectory tends to increase more over time due to the disparity between...
Fig. 12: SLAM results using DBEKF with dataset “B”, using only monocular observations from left camera (a), only stereo observations (b) and both monocular and stereo observations (c). Refer to figure 11 for graphical notation.

Fig. 13: Trace of the pose covariance from SLAM using DBEKF for different types of observations with dataset “A” (top) and “B” (bottom).

odometry measurements and the assumed scale from the observations.

As for stereo observations only experiments, it outperforms SLAM with monocular only in both datasets. Unfortunately, stereo observations seem to be not enough to cover the observations space, and as such some errors in pose estimation may still occur.

Finally, using both monocular and stereo observations, it performs better to stereo observations only by correcting further some trajectory errors. While with dataset “A” the pose covariance trace is always inferior to both monocular and stereo only experiments, with dataset “B” it is inferior to the monocular experiment but similar to the stereo experiment. The number of available features is superior when using both stereo and monocular observations, allowing for a better observation coverage when stereo observations are missing. Also, with this approach, one can use monocular features while suffering no scalability problem, proving to be advantageous over using monocular or stereo observations only.

C. DBEKF with different parameters

The utility weight factor $G$ can vary from 0 to 1. When $G = 0$, a landmark is immediately eliminated if for once it should be visible but is not detected. When $G = 1$ no landmark is removed regardless of its utility. However, the other two criteria are still applied, removing both landmarks with negative inverse depth and older landmarks when required.

Figures 14 and 15 present SLAM graphical results for dataset “A” and “B” with $G = 0$ and $G = 1$, respectively, while figure 16 shows the pose covariance evolution for those conditions. Regarding test results with $G = 0$, it is clear that the pose covariance rises significantly over time. This is due to the fact that it only takes one simple mismatch or occlusion to remove a landmark from the state, forcing the removal of landmarks that could otherwise be helpful for the state estimation. For test results with $G = 1$, results are near as good as with $G = 0.8$ and $T = 0.01$. However, it does not discard unwanted landmarks by their utility criteria from state, and if many of them are in state, they can risk the
VI. CONCLUSIONS

The usage of both cameras as a stereo vision decreases the uncertainty from all landmarks and allows a better initialization for the Simultaneous Localisation and Mapping (SLAM) algorithm, but the lack of stereo features may offer some problems to the SLAM problem if no other type of observations are used. With this work, both stereo and monocular features are used as observations, and as such one can use monocular information without worrying with the scalability problem as long as stereo observations are available. From the presented results, it is clear that using both monocular and stereo observations in the way introduced by this thesis increases the overall quality of SLAM over monocular only or stereo only observations, covering a bigger space of observations and suffering no lack of scale.

Although the usage of the Extended Kalman Filter (EKF) has been extensively used to solve the SLAM problem, its computational complexity grows with the number of landmarks to a point in time that it becomes unusable. This thesis showed that, with DBEKF, one can achieve good estimations with constant complexity when removing landmarks from state according to an evaluation criterion. It also became clear the need to estimate the utility of each landmark in regard with past evaluations.
REFERENCES


