

Making Visual SLAM Consistent with Geo-Referenced Landmarks

Guillaume Bresson*, Romuald Aufrère*[†] and Roland Chapuis*

*Institut Pascal - UMR 6602 CNRS – [†]LIMOS - UMR 6158 CNRS
Clermont Université, Université Blaise Pascal - Aubière, France
firstname.name@univ-bpclermont.fr

Abstract—This paper presents a solution to the consistency problem of SLAM algorithms. We propose here to model the drift affecting the estimation process. The divergence is seen as a bias on the vehicle localization. By using such a model, we are able to guarantee the consistency of the localization. We developed a filter taking into account the divergence and allowing to easily integrate any information helping to characterize the current drift. Geo-referenced landmarks are used in order to provide an absolute localization and drastically reduce the impact of the divergence. The filter is designed around an Extended Kalman Filter and is totally separated from the classical SLAM algorithm. Our method can consequently be connected to any existing SLAM process without trouble. A vehicle performing monocular SLAM in real time was used to validate our approach with real data. The results show that the integrity of the filter is preserved during the whole trajectory and that geo-referenced information helps reducing the natural SLAM drift.

I. INTRODUCTION

Intelligent vehicles are often associated to the fact of being autonomous. To reach this autonomy, vehicles need to be able to accurately localize themselves in their environment. Nevertheless, it is necessary to build cheap solutions usable in various environments in order for them to be viable. Likewise, relying on sensors affected by the context (low-cost GPS for instance) is not a good idea as it could lead to a wrong localization.

A widely used solution consists in incrementally building a map of the environment while localizing the vehicle in it [9]. Often called Simultaneous Localization And Mapping (SLAM), this method has been employed with an important diversity of algorithms and sensors [1]. Convergence properties have been demonstrated in a linear case [8] making SLAM a key element of vehicle autonomy. However, real life applications need more complex models which are essentially non-linear. In this context, SLAM-based methods drift over time [3] causing the filter to provide a biased vehicle localization and so wrong landmark estimates. Consistency (the true position is always located inside the estimated uncertainty) cannot be guaranteed in such cases [13]. The longer the trajectory is, the more important the drift becomes.

The inconsistency and inaccuracy of the localization furnished by SLAM algorithms have already been stated [6][14]. A small heading error from the vehicle can make the whole system diverge even more [2][11]. Anyway, with highly non-linear models, the linearization process can easily go wrong,

thus causing the inconsistency of the filter [12]. This phenomenon can be terrible in real life applications and must be taken into account in order to provide solutions suitable for long trajectories.

Several approaches have already been proposed to avoid the bias induced by the drift in the localization. In [10], for instance, the authors limit the size of the maps built. Each time linearization errors become too important, the submap is closed and a new one is started, with a new reference frame, freeing it from the drift of the previous map. However, the global map, linking all the submaps, is still affected by the drift. Other approaches usually rely on a loop closure to correct a part of the drift [16]. A dedicated algorithm is used in order to detect previously seen locations. By finding those locations, the drift can be estimated and reduced. This research is done outside of the SLAM process because of its inconsistency. Once a detection is made, the information is used inside the estimation algorithm to estimate the divergence. However, even when a loop is closed, the integrity of the solution cannot be certified. For example, landmarks located far from the loop closing point will not be as accurately estimated (they will still be affected by a drift) as those around the loop closure. Even if the drift is partially corrected, it does not mean that the solution is consistent.

Absolute information can also help estimating and reducing the drift. Indeed, an absolute localization is not subject to the divergence of the relative frame. However, it has already been stated that a low-cost GPS furnishes a global localization that can be affected by the surroundings. Geo-referenced infrastructures, on the other hand, could be used to communicate with vehicles in order to reduce the SLAM divergence [15]. The emergence of Vehicle To Infrastructure (V2I) communications, via the 802.11p standard, shows that collaborative approaches could be considered viable in a near future.

A more affordable solution at the time would be to have a geo-referenced map already available at the beginning of a trajectory. By finding common landmarks between the absolute map and the built map, the bias of the estimation process could be reduced. In [7], the authors integrate geo-referenced satellite images that are sought by a camera in order to localize precisely the robot. Building a database of geo-referenced landmarks or images, that can then be identified by the estimation algorithm, is a simple and efficient way to ensure

that the divergence will not grow too much.

However, avoiding that the drift becomes too important does not guarantee the integrity of the SLAM process. The bias affecting the vehicle localization and its evolution must be modeled in order to facilitate the integration of loop closing information or geo-referenced landmarks.

In this paper, we propose:

- A model to represent the divergence of SLAM algorithms
- A drift-aware filter, allowing the integration of geo-referenced information, that can be connected to any existing SLAM algorithm
- The application, in real time, of our method to a monocular SLAM with real experiments

The rest of the paper will be divided as follows: Section II will introduce the model developed to ensure the consistency of the estimation process. Section III will then detail the filter designed to integrate the drift. A special attention will be given to how geo-referenced landmarks can be incorporated inside the algorithm. Finally, the experiments and the obtained results will be detailed in Section IV.

II. DRIFT MODEL

The bias in the estimation process (induced by the SLAM drift) is evolving according to the distance traveled. As a consequence, the bias affecting the vehicle at one moment is different from the one impacting the same vehicle but several meters after. Similarly, the landmarks added to the map at different times are also subject to different biases. These considerations make necessary the construction of a dynamic model to truly represent the drift of SLAM processes.

It is possible to define the bias as the parameters allowing the transformation from a divergent frame (classic SLAM frame) to a drift-aware one. With a vehicle evolving in 3 dimensions (with the 3 corresponding angles), the bias can be defined as:

$$\mathbf{b} = (b_x \quad b_y \quad b_z \quad b_{\alpha_x} \quad b_{\alpha_y} \quad b_{\alpha_z})^T$$

The first three parameters are related to the position bias and last three ones to the angular drift. The estimated pose of a vehicle is defined the same way:

$$\mathbf{v} = (v_x \quad v_y \quad v_z \quad v_{\alpha_x} \quad v_{\alpha_y} \quad v_{\alpha_z})^T$$

Finally, a landmark i is represented in 3 dimensions:

$$\mathbf{l}_i = (l_{x_i} \quad l_{y_i} \quad l_{z_i})^T$$

Each of these 3 estimates is associated to a covariance matrix, respectively: \mathbf{P}_b , \mathbf{P}_v and \mathbf{P}_{l_i} . Moving from the biased frame to a consistent one is simple. For the vehicle, the bias is additive. Let be u the unbiased, drift-aware frame. The following relationship links the vehicle pose to its non-divergent counterpart:

$$\mathbf{v}_u = \begin{bmatrix} \mathbf{R}(b_{\alpha_z}, b_{\alpha_y}, b_{\alpha_x}) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \mathbf{v} + \mathbf{b} \quad (1)$$

where \mathbf{R} is the function computing the product of 2D rotation matrices based on the 3 angles.

For a landmark i , the relationship allowing to take into account the bias can be written as:

$$\mathbf{l}_{u_i} = \mathbf{R}(b_{\alpha_z}, b_{\alpha_y}, b_{\alpha_x}) \mathbf{l}_i + \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} \quad (2)$$

As mentioned before, the drift affecting the vehicle localization is not static. Its evolution is tightly bound to the distance traveled. This is why we chose to index the evolution of the dynamic bias on the curvilinear abscissa and not on the elapsed time since the beginning of a trajectory. It is more representative of the drift nature. Indeed, a stationary vehicle will not diverge more than it already has (as opposed to IMU errors which are cumulative). We thus define the evolution of the bias thanks to a random walk:

$$\mathbf{b}_s = \mathbf{b}_{s-1} + \boldsymbol{\varepsilon} \quad (3)$$

where s indices the current curvilinear abscissa and $s-1$ the one associated to the previous bias estimate. $\boldsymbol{\varepsilon}$ is a white noise representing the SLAM drift. This 6D-vector cannot be characterized. However, it is possible to cover all the realizations of this white noise thanks to a dedicated uncertainty. Each bias is associated to a covariance matrix allowing to represent the uncertainty about the drift estimation. Based on Equation (3), the covariance can be computed as follows:

$$\mathbf{P}_{b_s} = \mathbf{P}_{b_{s-1}} + \Delta s \mathbf{P}_{\boldsymbol{\varepsilon}} \quad (4)$$

with \mathbf{P}_b being the uncertainty (covariance matrix) of the bias, Δs the distance traveled between $s-1$ and s and $\mathbf{P}_{\boldsymbol{\varepsilon}}$ the uncertainty associated to the white noise $\boldsymbol{\varepsilon}$. $\mathbf{P}_{\boldsymbol{\varepsilon}}$ can be characterized experimentally based on the noticed quadratic divergence and the distance traveled. The process described here is stationary though it is likely that the drift grows faster during bends.

The uncertainty of the bias will grow linearly and will only decrease with an information allowing to estimate the drift itself. It means that, without a loop closing or an absolute information, the bias cannot be estimated but its uncertainty will ensure the consistency of the computed solution. Figure 1 shows an example of different realizations for a 1D bias and how it is still covered by its uncertainty.

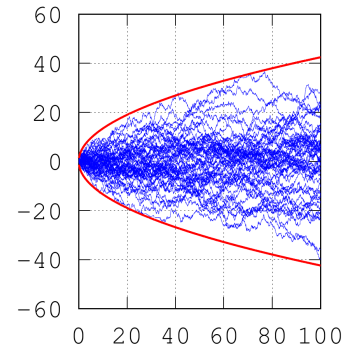


Fig. 1. Bias evolution according to the curvilinear abscissa (3σ uncertainty)

In this example, 50 different realizations of a unidimensional bias, with a maximum quadratic divergence of 20 units per meter, have been made according to the curvilinear abscissa (with steps $\Delta s = 0.1$). The blue curves are the different realizations. The red one represents the evolution of the bias standard deviation according to the curvilinear abscissa. It can be noticed that the drift is always between the bounds defined by the bias uncertainty.

III. DRIFT-AWARE FILTER

A. Filter architecture

Thanks to the model presented in Section II, it becomes possible to integrate the bias into the estimation process in order to take into account the drift induced by the SLAM algorithm. Each landmark and vehicle pose could be directly connected to the current bias inside the estimation process. However, by doing so, the filter would mix the uncertainties of the SLAM algorithm (error of the sensor measurement and vehicle uncertainty) and of the drift. As a result, the filter would be unable to separate the error caused by the divergence from the rest. It means that the whole uncertainty would be lowered by new landmark observations (the part related to the drift would be lowered but should not). To properly integrate the estimation of the drift in a SLAM process, it is necessary to use another filter, dedicated to this task.

This separation has also the advantage to allow any SLAM algorithm to be compatible with our method. Indeed, the algorithm presented in this paper only needs a vehicle pose, its uncertainty, landmark positions and their uncertainty. The “low-level” (classic) SLAM can use an Extended Kalman Filter or a Particle Filter, for instance, with any sensor (IMU, LRF, camera...). Thanks to this freedom, the drift-aware approach exposed here is not restricted to the application proposed in Section IV. Difficult environments or specific sensors can still be used with an algorithm specifically tailored for them. They can easily be plugged in the “high-level” (drift-aware) SLAM presented in this paper.

The low-level algorithm does not even need to perform SLAM. Indeed, as landmarks are then handled by the high-level process, there is no need to keep them at the local level when they are no longer of any interest. It becomes the task of the high-level SLAM. As a consequence, the low-level can be a fast localization algorithm using landmarks to help maintaining a proper vehicle pose.

It is worth noting that landmarks are sent to the high-level SLAM once and only when accurate (low uncertainty). Indeed, sending the same landmark regularly would require to take into account the bias each time a new estimation is available which can be costly. Moreover, having a landmark not accurate does not help the high-level SLAM to estimate the bias. The data association algorithm needed to find loop closures or identify geo-referenced landmarks will only work properly with precise landmarks. Once sent, landmarks will still be improved at the local level. However, the information gain will be low compared to the drift uncertainty, making this loss meaningless.

Figure 2 shows an example of the interactions between the two levels previously defined.

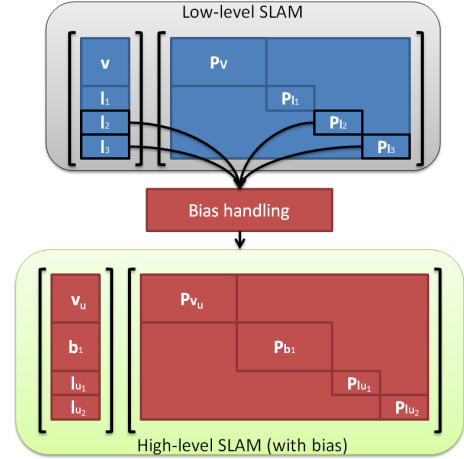


Fig. 2. Interactions between the drift-aware and the local SLAM

In the example depicted in Figure 2, two landmarks are sent to the high-level SLAM. When arriving, the current bias estimation (b_1 here) is applied to these landmarks. The cross-covariances (that are not sent from the low-level because they are negligible compared to the drift) can thus be estimated. Indeed, landmarks are almost only affected by the bias because their measurement uncertainties are very small. By applying the current bias estimate to landmarks, they will become naturally linked through the bias.

The two landmarks can then be deleted from the low-level SLAM. However, they will be kept as long as they are visible as they help maintaining the whole estimation process.

Figure 3 shows the state of the system after the convergence of another landmark.

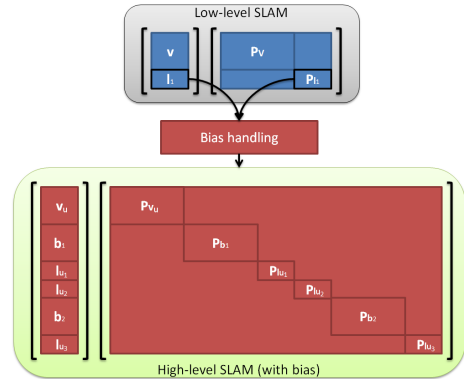


Fig. 3. Interactions between the drift-aware and the local SLAM after another landmark convergence

We can see that the low-level SLAM has deleted the previously sent landmarks, making it faster and lighter. On the other hand, the map of the SLAM with bias has grown. The new landmark has been connected to the rest of the state vector thanks to the bias. It can be noticed that a new bias estimate (b_2) has also been added to the map.

Two estimation of the bias are integrated into the map for 3 landmarks. However, the bias at a curvilinear abscissa s is different from the one at $s - 1$. It means that each landmark and vehicle pose should be linked to its own bias estimate. The problem is that adding a bias per landmark will increase the size of the map very quickly. Moreover, linking a new bias estimate with the rest of the state vector can become computationally expensive. We chose to integrate new bias estimates based on the distance traveled. This constraint is suitable as long as the distance between the initialization of new bias estimates is not too important.

Figure 4 illustrates how the whole algorithm is organized.

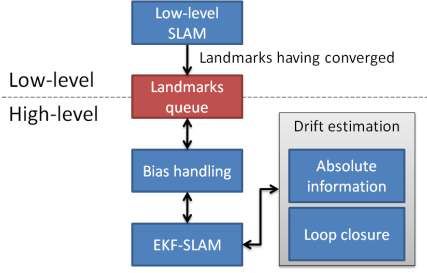


Fig. 4. Organization of the whole system

The two main parts of the algorithm (low-level and high-level) are only interacting through a queue of landmarks. Landmarks are copied in the queue once converged. The high-level, thanks to a timed thread, checks regularly if new landmarks are available. When it is the case, the current bias is taken into account before their integration into the map. It can be noticed that there is also a strong interaction between the drift-aware SLAM process and the algorithms able to estimate the bias. Though in this article we are interested in absolute measures (landmarks localized in the absolute frame), it is possible to consider other events that could provide more information about the bias, such as loop closures.

B. Drift integration and estimation

In order to achieve this degree of independence between the two levels, both are run by different threads. The low-level SLAM has priority over the rest as it is the one receiving information from the sensors. As mentioned in Figure 4, the high-level SLAM is handled by an Extended Kalman Filter (EKF). Its main advantage is to make easy the integration of data coming from different sensors. In our case, mixing the bias estimation with geo-referenced landmarks has been greatly simplified by the use of such a filter.

The EKF is also a great tool for initializations. It facilitates the connection of a new bias estimate to the rest of the state vector. In order to do so, the bias is first inserted into the state vector with an infinite variance. The EKF can then apply an update step to reduce the infinite variance towards the true uncertainty. By doing so, the links with the rest of the state vector will be created. Considering Equation (3), it is clear that \mathbf{b}_s cannot be estimated except with a loop closing or an absolute information. Consequently, the new bias estimate will

be the same as the previous one, but with a more important uncertainty.

The evolution of the variance is given by Equation (4). The difference between two consecutive bias estimates is used to refine the infinite uncertainty. We consider an observation function $h_b(\mathbf{b}_{s-1}, \mathbf{b}_s)$ operating the difference between those two biases. As stated before, their estimate is the same leading to an observation equals to zero. However, it is not the case for the variances which have evolved according to Equation (4). The observation error can thus be defined as:

$$\mathbf{R}_b = \mathbf{P}_{\mathbf{b}_s} - \mathbf{P}_{\mathbf{b}_{s-1}} \quad (5)$$

The Jacobian associated to the observation function h_b can then be computed:

$$\mathbf{H}_b = [\mathbf{0}_{6 \times \dots} \quad -\mathbf{I}_{6 \times 6} \quad \mathbf{0}_{6 \times \dots} \quad \mathbf{I}_{6 \times 6}] \quad (6)$$

where $-\mathbf{I}_{6 \times 6}$ corresponds to the derivative of the \mathbf{b}_{s-1} in the state vector and $\mathbf{I}_{6 \times 6}$ the derivative of the new bias estimate \mathbf{b}_s according to h_b . The two $\mathbf{0}_{6 \times \dots}$ matrices are the derivatives of the rest of the state vector (meaning landmarks and vehicle pose) which are not concerned by h_b .

It is then possible to compute the Kalman gain and perform an update step. Of course, the innovation equals zero as the drift cannot be directly estimated.

$$\mathbf{K} = \mathbf{P}\mathbf{H}_b^T (\mathbf{H}_b\mathbf{P}\mathbf{H}_b^T + \mathbf{R}_b)^{-1} \quad (7)$$

$$\mathbf{P} = \mathbf{P} - \mathbf{K}\mathbf{H}_b\mathbf{P} \quad (8)$$

where \mathbf{P} is the covariance matrix associated to the state vector of the high-level SLAM. After the update, the new bias estimate will be linked with the different landmarks and biases inside the map of the high-level SLAM.

Integrating a landmark inside the high-level SLAM shares some similarities with what has been explained before. The first step consists in the insertion of the landmark in the state vector with an infinite variance. Before that, the landmark must take into account the current bias estimate thanks to Equation (2). An update step of the EKF is then needed to refine the uncertainty. We thus define the observation \mathbf{z}_l (and its associated uncertainty \mathbf{R}_l) as the landmark estimate coming from the low-level SLAM:

$$\mathbf{z}_l = \mathbf{l}_i$$

$$\mathbf{R}_l = \mathbf{P}_{\mathbf{l}_i}$$

The observation function h_l is the non-linear function allowing to transform a drift-aware landmark to a classic one. It can be seen as the opposite of Equation (2):

$$h_l(\mathbf{b}_i, \mathbf{l}_{u_i}) = \mathbf{R}(b_{\alpha_{zi}}, b_{\alpha_{yi}}, b_{\alpha_{xi}})^T \left(\mathbf{l}_{u_i} - \begin{pmatrix} b_{xi} \\ b_{yi} \\ b_{zi} \end{pmatrix} \right) \quad (9)$$

It is then possible to compute \mathbf{H}_l , the Jacobian associated to h_l in order to update the uncertainty thanks to the Kalman gain. This way, the landmark uncertainty will integrate the bias error and will be bound to the rest of the map.

Thanks to Equations (6) and (9), the whole system is now aware of the drift. In order to correct it, loops in the trajectory

can be sought or absolute information can be integrated. In this paper, we chose to use accurate geo-referenced landmarks to increase the accuracy of the vehicle pose. A data association algorithm can be used to identify a geo-referenced landmark (in a database for example) based on its counterpart in the map of the high-level SLAM. For instance, in a monocular context, a landmark descriptor can be matched with those of a database in order to find a possible association. For a landmark \mathbf{l}_{u_j} already in the state vector of the high-level SLAM, an association is found with a geo-referenced landmark \mathbf{l}_{g_j} (g stands here for geo-referenced). The information can be integrated inside the EKF. To do so, the observation and its error are defined as follows (f stands for fusion):

$$\begin{aligned}\mathbf{z}_f &= \mathbf{l}_{g_j} \\ \mathbf{R}_f &= \mathbf{P}_{\mathbf{l}_{g_j}}\end{aligned}$$

The observation function h_f linking these two landmarks (\mathbf{l}_{u_j} and \mathbf{l}_{g_j}) is straightforward:

$$h_f(\mathbf{l}_{u_j}) = \mathbf{l}_{u_j} \quad (10)$$

This leads to the following innovation for the EKF:

$$\Delta = \mathbf{z}_f - h_f(\mathbf{l}_{u_j}) \quad (11)$$

Computing the Jacobian \mathbf{H}_f associated to h_f is very simple and will not be exposed here. As all the links between the landmarks have been created thanks to the bias estimates, the update performed by the EKF will naturally improve and precise the whole state vector.

IV. EXPERIMENTS

The validation of the drift-aware filter with geo-referenced landmarks presented in this paper requires the design of a low-level SLAM. We used a monocular EKF-SLAM [4][5] as cameras are cheap and provide information-rich data. The choice of a single camera facilitates the calibration phase that can be difficult otherwise. Odometry and steering angle are used in the prediction step of the EKF in the low-level. Odometric information allows to compensate the scale factor problem of monocular solutions by constraining it. Landmarks converging (uncertainty below a threshold) are sent to the high-level algorithm.

In order to simulate geo-referenced landmarks, a Real Time Kinematic GPS (RTK GPS), giving an accurate vehicle position, was used. The idea is to take a landmark which has just converged, pass it in the vehicle frame and then use the accurate vehicle pose given by the RTK GPS in order to transform the landmark to the absolute frame. It means that geo-referenced landmarks will still be affected by a small uncertainty here (the one after convergence). However, this error is small compared to the SLAM drift. As a consequence, the high-level SLAM will be improved with such an information. The objective of these experiments is to demonstrate the efficiency of our method. Using these simulated geo-referenced landmarks allow to achieve this goal.

The trajectory performed to validate our approach consists in long straight lines and important bends as they tend to

make the estimation process diverge a lot. An electrical vehicle (VIPALAB) was driven in an urban realistic platform (PAVIN). This platform is composed of various elements found in cities such as roads, buildings, roundabouts, sidewalks... The RTK GPS, employed to simulate geo-referenced landmarks, also served as a ground truth for comparison purposes. It helped evaluating the accuracy of our method. The camera, mounted on the ceiling of the vehicle, was running at 15 FPS. The vehicle was moving at approximately 2 meters per second. The developed application was running in real time.

48 landmarks were mapped (have converged) during the 100-meter trajectory. The convergence threshold has been set to 50 centimeters. A geo-referenced landmark is computed every 10 landmarks. It means that 4 geo-referenced landmarks have been fused during the trajectory. Figure 5 shows the divergence of the low-level SLAM and how it can still be consistent thanks to the bias integration.

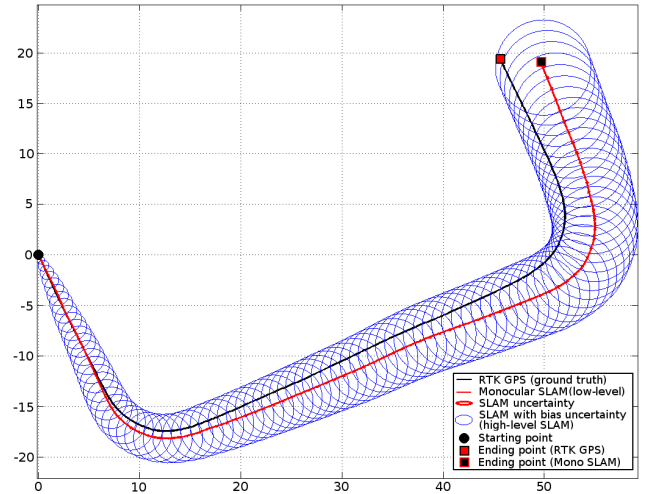


Fig. 5. Results of a drift-aware consistent monocular SLAM

Figure 5 illustrates the important gap between the true trajectory (black curve) and the one computed thanks to the monocular SLAM (red curve). It can be noticed that the drift is not only a position bias but also an angular one. The very small uncertainty of the SLAM algorithm (red ellipses) clearly demonstrates that the integrity of the filter is lost after only a few meters. However, by taking into account the bias (blue ellipses), the consistency is kept all along the trajectory. The results when using geo-referenced landmarks to correct the drift are visible in Figure 6.

We can see that, with only 4 geo-referenced landmarks, the drift is well-corrected (blue curve). Each time an absolute information is integrated, the uncertainty of the vehicle pose is drastically reduced (blue ellipses). Even when the divergence is not fully corrected, the consistency of the high-level filter is preserved. The small jumps in the trajectory are caused by the initialization of a new bias estimate. This can easily be smoothed by the use of more bias estimates. Of course, integrating more geo-referenced landmarks would also improve the filter accuracy. However, these results show that, with only

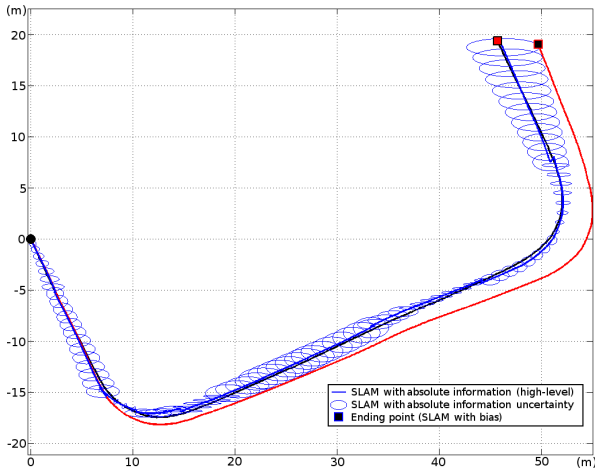


Fig. 6. Results of a drift-aware SLAM using geo-referenced landmarks

a few absolute landmarks, an important part of the divergence is corrected. Anyway, it can be noticed that consistency is maintained during the whole trajectory. Figure 7 shows the lateral and longitudinal errors of the high-level SLAM after the integration of geo-referenced landmarks.

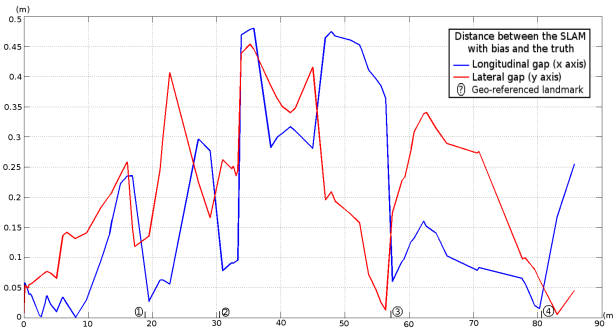


Fig. 7. Distance between the computed vehicle position and the truth all along the trajectory

The average longitudinal and lateral errors exposed in Figure 7 are around 20 centimeters. It can be noticed that the error is always less than 50 centimeters. It corresponds to the convergence threshold used for the computation of the simulated geo-referenced landmarks. It means that by using true absolute landmarks, the error could be lowered as they would not be impacted by the small convergence uncertainty. As already mentioned, with more geo-referenced landmarks, the vehicle pose would be improved.

V. CONCLUSION

In this paper, a solution to model the natural divergence of SLAM processes has been presented. Thanks to it, we are able to guarantee the consistency of the vehicle pose. To do so, a dedicated algorithm, based on an Extended Kalman Filter, was designed. Its role is to take into account the drift, which we chose to see as a bias of the estimation process.

An autoregressive model of order 1 is used to model the current bias and its evolution. By integrating it into the EKF,

we are able to link the whole map to its associated bias. The integrity of the estimation process is thus preserved.

This special architecture separates the bias estimation from the usual SLAM. The direct consequence is that any existing SLAM process (with any sensor or method) can be connected to the bias estimation. It only needs to provide landmarks and their uncertainty to the dedicated filter.

We applied our approach to a monocular context. The experiments, running in real time in an urban environment, illustrate that the consistency of the filter is kept all along the trajectory performed. Geo-referenced landmarks were integrated in order to correct the divergence. The results show that it is a good solution to reduce the estimation errors bound to the SLAM divergence.

In future work, we plan to integrate the notion of bias presented here in a decentralized algorithm in order to fuse data coming from different vehicles.

REFERENCES

- [1] T. Bailey and H. Durrant-Whyte. Simultaneous Localization and Mapping (SLAM): Part II. *IEEE Robotics and Automation Magazine*, 13(3):108–117, 2006.
- [2] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. Consistency of the EKF-SLAM Algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562–3568, 2006.
- [3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, 2001.
- [4] G. Bresson, T. Féraud, R. Aufrère, P. Checchin, and R. Chapuis. A New Strategy for Feature Initialization in Visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Perception and Navigation for Autonomous Vehicles in Human Environment*, pages 115–120, 2011.
- [5] G. Bresson, T. Féraud, R. Aufrère, P. Checchin, and R. Chapuis. Parsimonious Real Time Monocular SLAM. In *IEEE International Conference on Intelligent Vehicles*, pages 511–516, 2012.
- [6] J. A. Castellanos, J. Neira, and J. D. Tardós. Limits to the Consistency of EKF-Based SLAM. In *5th IFAC Symposium on Intelligent Autonomous Vehicles*, 2004.
- [7] G. Conte and P. Doherty. Vision-Based Unmanned Aerial Vehicle Navigation Using Geo-Referenced Information. *EURASIP Journal On Advances In Signal Processing*, pages 10–32, 2009.
- [8] M. W. M. G. Dissanayake, P. Newman, H. F. Durrant-Whyte, S. Clark, and M. Csorba. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.
- [9] H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. *IEEE Robotics and Automation Magazine*, 13(2):99–110, 2006.
- [10] C. Estrada, J. Neira, and J. D. Tardós. Hierarchical SLAM: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596, 2005.
- [11] U. Frese. A Discussion of Simultaneous Localization and Mapping. *Autonomous Robots*, 20(1):25–42, 2006.
- [12] S. Huang and G. Dissanayake. Convergence and Consistency Analysis for Extended Kalman Filter Based SLAM. *IEEE Transactions on Robotics*, 23(5):1036–1049, 2007.
- [13] S. Julier and J. Uhlmann. Building a Million Beacon Map. In *Sensor Fusion and Decentralized Control in Robotic Systems IV*, volume 4571, pages 1–9, 2001.
- [14] A. Martinelli, N. Tomatis, and R. Siegwart. Some Results on SLAM and the Closing the Loop Problem. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2917–2922, 2005.
- [15] D.-S. Seo, D. Won, G.-W. Yang, M.-S. Choi, S.-J. Kwon, and J. W. Park. A Probabilistic Approach for Mobile Robot Localization under RFID Tag Infrastructures. In *International Conference on Control, Automation and Systems*, pages 1797–1801, 2005.
- [16] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós. A comparison of loop closing techniques in monocular SLAM. *Robotics and Autonomous Systems*, 57(12):1188–1197, 2009.