

A New Strategy for Feature Initialization in Visual SLAM

Guillaume Bresson, Thomas Féraud, Romuald Aufrère, Paul Checchin,
Roland Chapuis

► **To cite this version:**

Guillaume Bresson, Thomas Féraud, Romuald Aufrère, Paul Checchin, Roland Chapuis. A New Strategy for Feature Initialization in Visual SLAM. IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Perception and Navigation for Autonomous Vehicles in Human Environment, 2011, San Francisco, United States. <hal-01351394>

HAL Id: hal-01351394

<https://hal.inria.fr/hal-01351394>

Submitted on 3 Aug 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A New Strategy for Feature Initialization in Visual SLAM

Guillaume Bresson*, Thomas Féraud*, Romuald Aufrère*[†], Paul Checchin* and Roland Chapuis*

*LASMEA - UMR 6602 CNRS – [†]LIMOS - UMR 6158 CNRS

Blaise Pascal University

Aubière, France

first-name.last-name@univ-bpclermont.fr

Abstract—This paper presents a Visual EKF-SLAM process using an original and very efficient strategy for initializing landmarks. Usually, with Cartesian coordinates, new points are created along the line-of-sight with a large variance. However, this type of initialization is subject to significant linearization issues making landmarks diverge from their real position. The immediate consequence is a failure of the Visual SLAM process. We propose here a new strategy that avoids or drastically limits the linearization errors. The first part of this strategy takes place during the tracking process where a coherent window is needed in order to successfully follow a point and make it converge. The second part concerns the update step. Due to linearization errors, a landmark in front of the observer can be updated behind it. We compute a corrective of the Kalman gain in order to preserve the integrity. We applied this strategy to real data illustrating its efficiency.

I. INTRODUCTION

The Simultaneous Localization And Mapping (SLAM) has received a lot of attention over the last two decades [10]. Though solutions to the main problems have been found [9], they mostly rely on the use of range and bearing sensors [2]. However, sensors like LRFs (Laser Range Finders) are expensive and usually only allow 2D perception. Conversely, monocular SLAM is cheap and allows to estimate the 6 DoF of a camera pose (3D position and the 3 associated angles). Nevertheless, cameras are bearing-only sensors. Estimating the accurate depth of a landmark requires many observations with a sufficient parallax. When dealing with Visual SLAM, it is thus a difficult problem to correctly initialize and track landmarks until they converge to their real position.

There exist various ways to perform SLAM. Among them, the most popular are Extended Kalman Filters (EKFs) [20], particle filters [11][22], bundle adjustment based methods [14] and Unscented Kalman Filters (UKFs) [5]. All these approaches can be expensive in terms of memory and processing time. Nevertheless, we chose the EKF as submapping methods [4][12] can be utilized in order to limit this load.

When dealing with bearing-only SLAM, two types of initialization can be found in the literature: delayed and undelayed ones. Delayed initialization, like in [1] and [8], has the advantage to only use accurately defined landmarks. However, it means that a large amount of information is lost. For example, when the vehicle is turning, landmarks will pass out of the field-of-view without being initialized leading to a large error on the localization of the robot. Concerning undelayed initializations, the easiest way to perform it would

be to create a point on the line-of-sight of its first observation with a large variance. However, this leads to significant linearization issues when tracking and updating landmarks [3]. To avoid this problem, the Inverse Depth (ID) parametrization is usually chosen to represent landmarks [7][16]. Yet, this solution is an over-parametrization of the problem where 6 parameters are required to estimate a 3D pose. Consequently, this representation is computationally expensive (size of the state vector duplicated) and so not suitable for long distance SLAM. In practice, a strategy allowing to switch from ID parametrization to Cartesian coordinates is used to prevent the computational load from becoming too significant [6]. In [15] and [21], multiple hypothesis are utilized for a single point. They are integrated into the state vector and discarded when considered wrong. This solution is also computationally expensive and needs a special strategy when performing the update step of the Kalman filter due to the wrong hypothesis inserted in the state vector.

In this paper, we propose a solution to the problems induced by the linearization steps when using a simple Cartesian representation for landmarks in the EKF framework. The problem we address here is twofold:

- Getting a coherent tracking window (uncertainty projection into the image plane) from a landmark when the real point is far from the fictitious one
- Updating correctly the state vector when the landmark is initialized far from its real position

The next section will show why tracking and update fail when using the classical Jacobian computation. Section III will present a solution to the first problem stated (tracking window). Then, in Section IV, a correction of the Kalman gain is proposed so as to avoid the loss of the integrity. Next, Section V will present the experiments conducted and the results obtained illustrating that our approach is efficient. Finally, Section VI will conclude and give some insights about what we are planning to do next.

II. PROBLEM STATEMENT

Considering an observation in the image plane $\mathbf{z}_i = (u \ v)^T$, we search the 3D point corresponding in the camera frame $\mathbf{z}_c = (x \ y \ z)^T$. A representation of the frames used in this paper is given in Figure 1 (the mobile camera evolves along the \vec{x} axis).

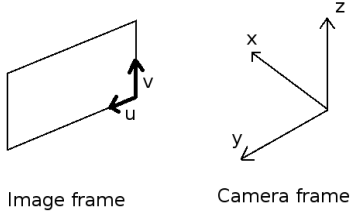


Fig. 1. Camera and image frames used in this paper.

The pinhole model provides the following relation:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \mathbf{F} \cdot \mathbf{z}_c \quad (1)$$

with

$$\mathbf{F} = \begin{pmatrix} c_u & f_u & 0 \\ c_v & 0 & f_v \\ 1 & 0 & 0 \end{pmatrix}$$

where f_u and f_v are the focal distances in pixels and c_u and c_v represent the coordinates of the optical center in the image frame. We initialize the landmark with a fictive depth (x coordinate) corresponding to half the maximum distance possible for a point. This distance has to be chosen depending on the environment the camera is moving in. Thanks to this fictive depth, we can infer the pose of the fictitious 3D point:

$$\begin{cases} y = \frac{ux - c_u x}{f_u} \\ z = \frac{vx - c_v x}{f_v} \end{cases} \quad (2)$$

The covariance associated to this point also needs to be initialized. Let $g(\mathbf{z}_c, \mathbf{F})$ be the non-linear function linking the 3D point to the 2D one. Let \mathbf{G} be the Jacobian associated to g . This gives the following relation:

$$\mathbf{P}_i = \mathbf{G} \mathbf{P}_c \mathbf{G}^T \quad (3)$$

where \mathbf{P}_i and \mathbf{P}_c are respectively the covariance of the point newly initialized in the image and the camera frame:

$$\mathbf{P}_i = \begin{pmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{uv} & \sigma_v^2 \end{pmatrix} \quad \mathbf{P}_c = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 \end{pmatrix}$$

and \mathbf{G} is defined as follows:

$$\mathbf{G} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \end{pmatrix}$$

with

$$\frac{\partial u}{\partial x} = -\frac{f_u y}{x^2} \quad \frac{\partial u}{\partial y} = \frac{f_u}{x} \quad \frac{\partial u}{\partial z} = 0$$

$$\frac{\partial v}{\partial x} = -\frac{f_v z}{x^2} \quad \frac{\partial v}{\partial y} = 0 \quad \frac{\partial v}{\partial z} = \frac{f_v}{x}$$

The error made on a new point cannot be greater than the observation noise (defined as a diagonal matrix) as no updates (or other observations) have been made yet. To initialize properly the covariance of the 3D point, we rotate it on the \vec{x} axis of the camera frame. This leads to the following 3D point: $(\rho \ 0 \ 0)^T$ where $\rho = \sqrt{x^2 + y^2 + z^2}$ and consequently the following diagonal covariance:

$$\mathbf{P}'_c = \begin{pmatrix} (\rho - d_{min})^2 & 0 & 0 \\ 0 & \frac{\sigma_u^2 \rho^2}{f_u^2} & 0 \\ 0 & 0 & \frac{\sigma_v^2 \rho^2}{f_v^2} \end{pmatrix} \quad (4)$$

where d_{min} is the minimal distance from which the camera can see. \mathbf{P}'_c is then put on the line-of-sight of the observation thanks to the appropriate rotation. Finally, the estimated pose of the camera allows to put the point and its covariance in the world frame. The uncertainty associated to the position of the camera is taken into account during this step.

The point fully initialized is then tracked in order to make it converge towards its real position. Let $\hat{\mathbf{x}}$ be the state vector (including the camera and landmark poses). Let $h(\hat{\mathbf{x}}_{k|k-1})$ be the non-linear function (observation function) utilized to predict the measurement from the predicted state $\hat{\mathbf{x}}_{k|k-1}$ at time k and \mathbf{H}_k be the Jacobian associated to h . Let $\mathbf{H}_{c,k}$ be the Jacobian associated to the covariance of the mobile sensor $\mathbf{P}_{c,k|k-1}$ and $\mathbf{H}_{l_w,k}$ be the Jacobian associated to the covariance of the landmark l in the world frame. The prediction of the covariance of a landmark l in the image frame can be computed as follows:

$$\mathbf{P}_{l_i,k|k-1} = \mathbf{H}_{c,k} \mathbf{P}_{c,k|k-1} \mathbf{H}_{c,k}^T + \mathbf{H}_{l_w,k} \mathbf{P}_{l_w,k|k-1} \mathbf{H}_{l_w,k}^T \quad (5)$$

However, in (5), we linearize around the fictitious point which can be far from the real pose causing the projection of the ellipsoid corresponding to the covariance matrix to be wrong. The tracking issues resulting from this are illustrated in Figure 2 where no updates are performed (prediction only).

Section III will present a method to compute a bounding box that properly fits the projection of the ellipsoid into the image.

Another problem arises when performing the update step of the EKF. The equations of the update are as follows:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})) \quad (6)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \quad (7)$$

where \mathbf{z}_k is the observation at time k and \mathbf{K}_k is the Kalman gain whose equation is:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (8)$$

where \mathbf{R}_k is the observation noise. The problem is similar to the one exposed previously. Indeed, we linearize around the

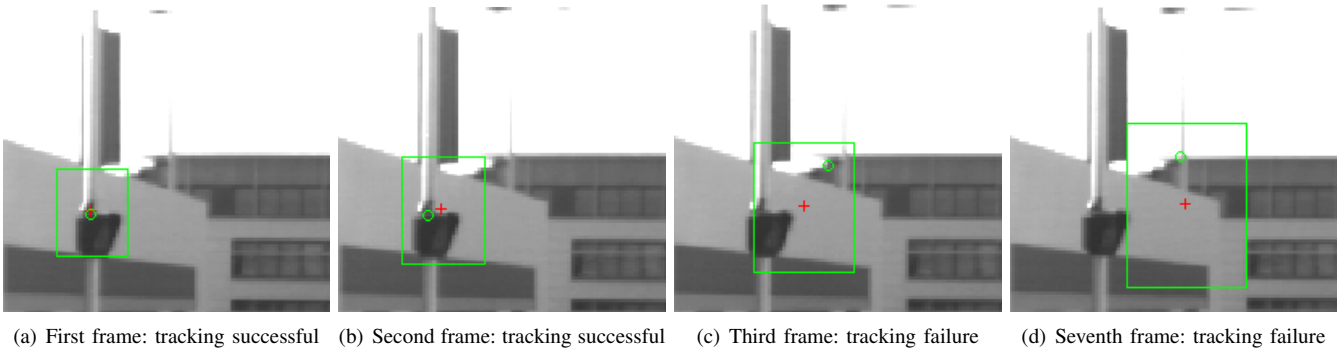


Fig. 2. Tracking failure due to significant linearization errors. The green rectangle is the predicted window where the landmark is supposed to be. The red cross is the estimate of the landmark position. The green circle is the observation (patch matched with ZNCC).

fictitious point leading to a wrong estimate. As a consequence, a landmark can be updated at a position behind the vehicle which is impossible. This problem is illustrated in Figure 3.

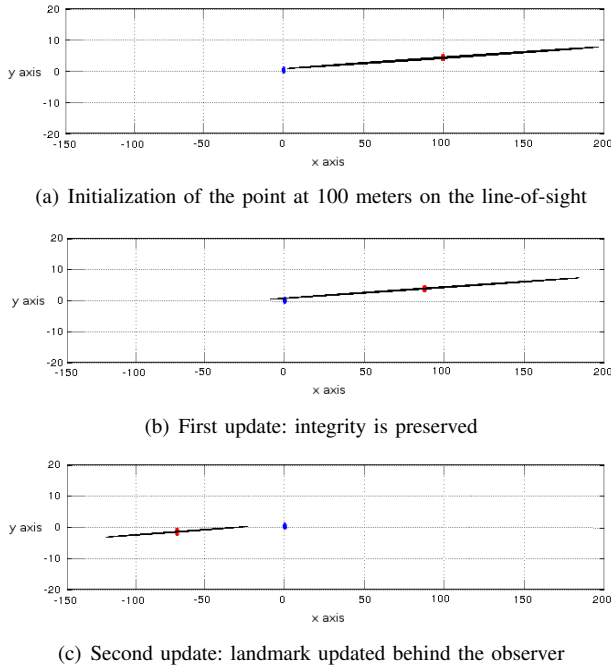


Fig. 3. Top view of a point updated behind the observer. The blue circle is the position of the vehicle. The red square is the landmark. The black ellipse is the uncertainty associated to the landmark.

This problem has also been stated and treated for ID parametrization in [17] but the proposed solution cannot be applied with Cartesian landmarks. In Section IV, we will present a solution working with a simple Cartesian pose that will correct the update step and prevent the landmarks from being updated at a wrong position.

III. BOUNDING BOX FOR TRACKING

We propose here a method to find the bounding box corresponding to the projection of the ellipsoid associated to a landmark uncertainty in the image frame thanks to its tangent planes. Other methods exist to find a proper projection. In

[18], Laplace's method is used to correctly fit a Gaussian at the global maximum of the observation function. However, a minimization step is required to find a correct solution. The approach proposed here does not rely on a minimization step. The idea is to find the planes tangent to the ellipsoid that give the maximum size of the corresponding ellipse in the image. Four tangent planes are needed in order to have a correct bounding box. Let \mathbf{P}_c be the covariance matrix of a landmark in the camera frame:

$$\mathbf{P}_c = \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix}$$

and

$$\begin{aligned} \mathbf{P}_c^{-1} &= \frac{1}{\det \mathbf{P}_c} \begin{pmatrix} df - e^2 & ce - bf & be - cd \\ ce - bf & af - c^2 & cb - ae \\ be - cd & cb - ae & ad - b^2 \end{pmatrix} \\ &= \begin{pmatrix} A & B & C \\ B & D & E \\ C & E & F \end{pmatrix} \end{aligned}$$

Then, a relationship verified by the points on the surface of the ellipsoid \mathcal{E} generated by \mathbf{P}_c is the Mahalanobis distance:

$$\begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T \mathbf{P}_c^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \quad (9)$$

where $(x_0 \ y_0 \ z_0)^T$ is the center of \mathcal{E} . The planes tangent to \mathcal{E} are also orthogonal to the surface normal. Moreover, the normal at a point on the surface is given by the gradient:

$$\begin{aligned} \vec{n} &= \nabla \mathcal{E}(x, y, z) = \left(\frac{\partial \mathcal{E}}{\partial x} \quad \frac{\partial \mathcal{E}}{\partial y} \quad \frac{\partial \mathcal{E}}{\partial z} \right)^T \\ &= \begin{pmatrix} 2(x - x_0)A + 2(y - y_0)B + 2(z - z_0)C \\ 2(x - x_0)B + 2(y - y_0)D + 2(z - z_0)E \\ 2(x - x_0)C + 2(y - y_0)E + 2(z - z_0)F \end{pmatrix} \end{aligned} \quad (10)$$

To be part of a plane p , orthogonal to the normal, a point must verify the following relation:

$$\begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} \in p \Leftrightarrow \begin{pmatrix} x_p - x \\ y_p - y \\ z_p - z \end{pmatrix}^T \cdot \vec{n} = 0 \quad (11)$$

As we are looking for the points located on the planes passing by the origin (camera), we can simplify equation (11):

$$(x \ y \ z) \cdot \vec{n} = 0 \quad (12)$$

The other constraint we can add is that the tangent planes intersect the image plane to form vertical or horizontal lines. This means that vertical planes will contain the \vec{z} axis and horizontal ones will contain the \vec{y} axis. Consequently, for the points belonging to the vertical planes:

$$\vec{z} \cdot \vec{n} = 0 \Leftrightarrow (x - x_0)C + (y - y_0)E + (z - z_0)F = 0 \quad (13)$$

and for those belonging to the horizontal planes:

$$\vec{y} \cdot \vec{n} = 0 \Leftrightarrow (x - x_0)B + (y - y_0)D + (z - z_0)E = 0 \quad (14)$$

These constraints give us the following system to solve for the vertical planes:

$$\begin{cases} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T \mathbf{P}_c^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \\ (x \ y \ z) \cdot \vec{n} = 0 \\ (x - x_0)C + (y - y_0)E + (z - z_0)F = 0 \end{cases} \quad (15)$$

and the following for the horizontal planes:

$$\begin{cases} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T \mathbf{P}_c^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \\ (x \ y \ z) \cdot \vec{n} = 0 \\ (x - x_0)B + (y - y_0)D + (z - z_0)E = 0 \end{cases} \quad (16)$$

By solving the system (15), we find two points on the ellipsoid matching the different constraints. We then project them into the image thanks to the intrinsic parameters. This gives us the left and right limits of the bounding box. We also find two points by resolving (16) which gives us the top and bottom limits of the bounding box. Thanks to this information, we can successfully track the landmarks in the image plane. Figure 4 gives, with the same set of images used in Figure 2, the tracking results with the bounding box method presented here. The landmark is tracked successfully with no update steps (prediction only) illustrating the effectiveness of our approach.

IV. UPDATE CORRECTION

When dealing with a point newly initialized, the Jacobian associated to the observation function h can be erroneous as it is linearized around a fictitious point which can be far from the real landmark pose (cf. Fig. 3).

In order to avoid this problem, we introduce here a corrective of the Kalman gain [13] whose goal is to limit the impact of the state update on the point when it is needed. Indeed, as long as the update does not put the landmark before the observation, the Kalman gain is used without any correction. Otherwise, the idea is to have the following relation verified after the update:

$$\mathbf{z}_k = h(\hat{\mathbf{x}}_{k|k}) \quad (17)$$

Indeed, as the transformation from one space (image) to another of higher dimension (camera) is not well-conditioned, it is necessary to find a factor that makes (17) true. Let Ω_k be the Kalman gain once the corrective factor r has been applied. It is thus defined as follows:

$$\Omega_k = r \times \mathbf{K}_k \quad (18)$$

In the case of our Visual SLAM, for a landmark $\mathbf{x}_l = (x_l \ y_l \ z_l)^T$ defined in the world frame, the observation function can be written:

$$\begin{cases} u_{est} = \frac{\mathbf{F}_1 \mathbf{R}_{cw}^T (\mathbf{x}_l - \mathbf{t}_{cw})}{\mathbf{F}_3 \mathbf{R}_{cw}^T (\mathbf{x}_l - \mathbf{t}_{cw})} \\ v_{est} = \frac{\mathbf{F}_2 \mathbf{R}_{cw}^T (\mathbf{x}_l - \mathbf{t}_{cw})}{\mathbf{F}_3 \mathbf{R}_{cw}^T (\mathbf{x}_l - \mathbf{t}_{cw})} \end{cases} \quad (19)$$

where u_{est} and v_{est} correspond to the estimated position of the landmark in the image, \mathbf{F}_i is the i^{th} line of the intrinsic parameters matrix, \mathbf{R}_{cw} is the rotation matrix passing points from the camera frame to the world frame and \mathbf{t}_{cw} is the translation associated to the rotation. With $\mathbf{z}_k = (u_{obs} \ v_{obs})^T$ and by taking into account (17) and (18), we can write:

$$\begin{cases} u_{obs} = \frac{\mathbf{F}_1 \mathbf{R}_{cw}^T (\mathbf{x}_l + \Omega \Delta - \mathbf{t}_{cw})}{\mathbf{F}_3 \mathbf{R}_{cw}^T (\mathbf{x}_l + \Omega \Delta - \mathbf{t}_{cw})} \\ v_{obs} = \frac{\mathbf{F}_2 \mathbf{R}_{cw}^T (\mathbf{x}_l + \Omega \Delta - \mathbf{t}_{cw})}{\mathbf{F}_3 \mathbf{R}_{cw}^T (\mathbf{x}_l + \Omega \Delta - \mathbf{t}_{cw})} \end{cases} \quad (20)$$

with the innovation $\Delta = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})$ and Ω the corrected Kalman gain associated to \mathbf{x}_l . This leads to the two following corrections:

$$r_u = \frac{(u_{obs} - u_{est})De}{(\mathbf{F}_1 - u_{obs}\mathbf{F}_3)\mathbf{R}_{cw}^T \mathbf{K} \Delta} \quad (21)$$

$$r_v = \frac{(v_{obs} - v_{est})De}{(\mathbf{F}_2 - v_{obs}\mathbf{F}_3)\mathbf{R}_{cw}^T \mathbf{K} \Delta} \quad (22)$$

where $De = \mathbf{F}_3 \mathbf{R}_{cw}^T (\mathbf{x}_l - \mathbf{t}_{cw})$. As these scalars are correctors they cannot exceed 1. Keeping a corrective factor above 1 will cause the EKF to be overconfident leading to a wrong estimate. We will keep the lowest value from (21) and (22) to avoid this problem. If the chosen corrector is greater than 1, then no correction is applied.

This corrective factor will prevent the landmarks from being updated behind the observer. Linearization errors are drastically reduced and the integrity is maintained. More details

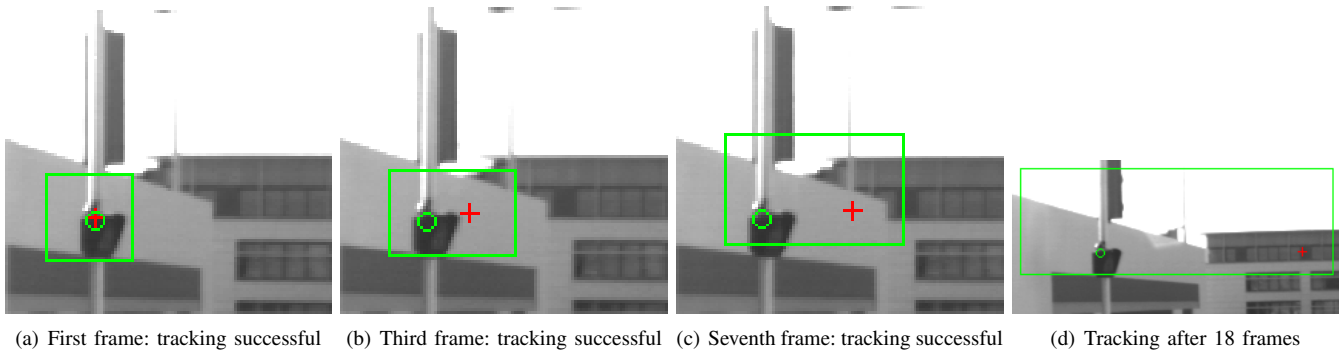
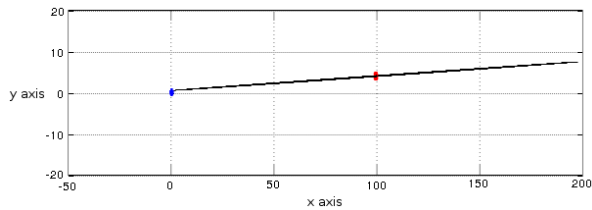
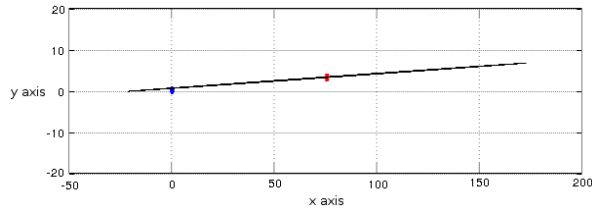


Fig. 4. Tracking with the bounding box method presented here. The green rectangle is the predicted window where the landmark is supposed to be. The red cross is the estimate of the landmark position. The green circle is the observation (patch matched with ZNCC).

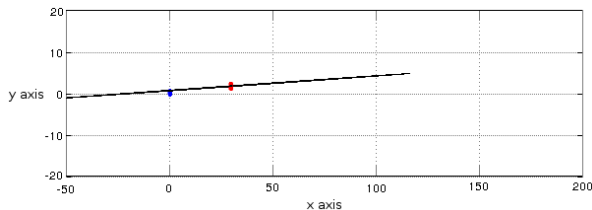
about this correction can be found in [13]. Figure 5 shows the same update case as Figure 3 with the corrective factor applied to the Kalman gain. The updated position of the landmark is coherent and converging towards its real one.



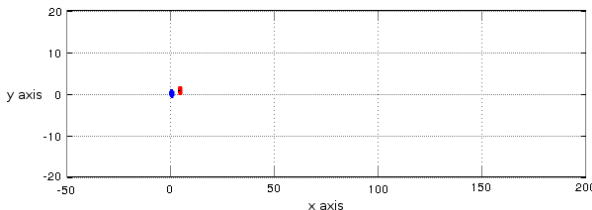
(a) Initialization of the point at 100 meters on the line-of-sight



(b) First update: integrity is preserved



(c) Second update: integrity still preserved



(d) After 10 updates, the landmark has converged

Fig. 5. Top view of a properly updated point thanks to the Kalman gain correction. The blue circle is the position of the vehicle. The red square is the landmark. The black ellipse is the uncertainty associated to the landmark.

V. EXPERIMENTS

In order to evaluate the efficiency of our monocular SLAM, a 30-meter trajectory has been performed. It was accomplished with an electrical vehicle (Cycab) in an small realistic urban experimental platform called PAVIN. This platform recreates a human environment with paved roads, crosswalks, traffic lights, building façades... Our vehicle was equipped with a single camera and proprioceptive sensors giving odometric information and the steering angle. A Differential GPS was also available but was only used as the ground truth for comparison purposes. The camera was mounted on top of the vehicle (2 meters) and acquired 15 images per second. The vehicle was traveling at approximately 1 meter per second. Landmarks were initialized by extracting areas of 15×15 pixels around Shi-Tomasi features [19]. Data association was performed using Normalized Cross Correlation between the previously extracted patch and the computed bounding box.

The trajectory was used to compare two algorithms: a naive implementation of a Visual EKF-SLAM with a simple initialization and a similar version with the strategy presented in this paper. This trajectory is composed of a long straight line to show that landmarks are tracked and updated correctly over a long distance and of a bend to demonstrate that the localization remains consistent with landmarks passing out of view quickly. Figure 6 shows the results of the two algorithms with at least 10 landmarks available per image.

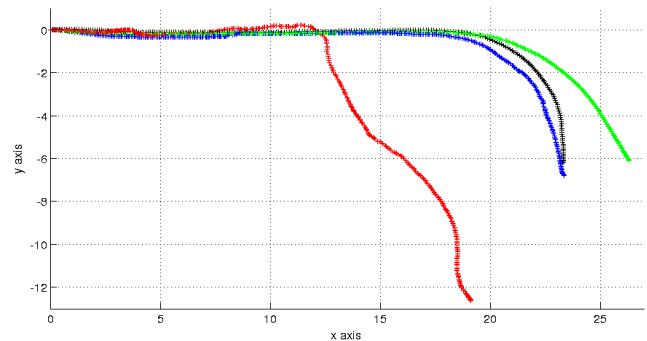


Fig. 6. Localization of the vehicle. **Black**: ground truth (DGPS). **Green**: prediction only (based on proprioceptive sensors). **Red**: naive EKF-SLAM. **Blue**: our strategy applied to the EKF-SLAM.

As expected, the naive implementation diverges completely whereas our approach remains consistent during the whole trajectory. Nevertheless, the restrictions on the size of the bounding box and the slow velocity of the vehicle have allowed the naive SLAM to be correct during the beginning of the trajectory. Thanks to our strategy, the bend has been followed quite closely even though the proprioceptive information was not accurate. These results can be improved with a smarter data association and more trusted landmarks especially when proprioceptive sensors are not reliable. The number of landmarks initialized is a good indicator of the quality of a monocular SLAM as new landmarks are only created when old ones cannot be matched (mostly due to linearization errors). Table I indicates the number of landmarks that have been initialized and that have converged during the whole trajectory.

| | Naive SLAM | Our SLAM |
|---------------------------------|------------|----------|
| Number of landmarks initialized | 650 | 182 |
| Number of landmarks conserved | 79 | 61 |

TABLE I
NUMBER OF INITIALIZATIONS.

We can see that far less landmarks are initialized with our method. Indeed, tracking is possible as long as a landmark is visible and the update step is controlled by the corrective factor applied to the Kalman gain. As a consequence, landmarks converge within a few frames and are still being predicted correctly. However, an important number of landmarks are eliminated during the process. For most of them, they have disappeared of the field-of-view before convergence. For the others, a better data association could avoid this problem.

VI. CONCLUSION

We have presented a Visual EKF-SLAM that is able to use landmarks defined in Cartesian coordinates. The significant linearization errors occurring during the tracking and update steps are avoided or reduced thanks to the strategy exposed in this paper.

Our approach consists of a way to easily compute a bounding box that allows to track a landmark and make it converge towards its real position. This method, based on the computation of tangent planes, permits to find a bounding box which is coherent with the 3D uncertainty of a landmark.

The second part of this strategy permits to drastically reduce linearization errors during the update step. It is done thanks to the computation of a corrective factor able to reduce the impact of the update on a landmark. It allows to preserve the integrity when updating a landmark position.

We have evaluated our system with real data showing that our approach is efficient. We have compared our results with a naive implementation of a monocular EKF-SLAM and demonstrated that our strategy allows a more accurate localization while requiring less landmark initializations.

For future work, we plan to decentralize our monocular EKF-SLAM over several vehicles in order to have a better localization accuracy.

REFERENCES

- [1] T. Bailey. Constrained Initialisation for Bearing-Only SLAM. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1966–1971, 2003.
- [2] T. Bailey and H. Durrant-Whyte. Simultaneous Localization and Mapping (SLAM): Part II. *IEEE Robotics and Automation Magazine*, 13(3):108–117, 2006.
- [3] K. E. Bekris, M. Glick, and L. E. Kavraki. Evaluation of Algorithms for Bearing-Only SLAM. In *IEEE International Conference on Robotics and Automation*, pages 1937–1943, 2006.
- [4] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An Atlas Framework for Scalable Mapping. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1899–1906, 2003.
- [5] D. Checklov, M. Pupilli, W. Mayol-Cuevas, and A. Calway. Real-Time and Robust Monocular SLAM using Predictive Multi-Resolution Descriptors. *Advances in Visual Computing*, 4292:276–285, 2006.
- [6] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse Depth to Depth Conversion for Monocular SLAM. In *IEEE International Conference on Robotics and Automation*, pages 2778–2783, 2007.
- [7] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, 2008.
- [8] A. J. Davison. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *IEEE International Conference on Computer Vision*, pages 1403–1410, 2003.
- [9] 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.
- [10] H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. *IEEE Robotics and Automation Magazine*, 13(2):99–110, 2006.
- [11] E. Eade and T. Drummond. Scalable Monocular SLAM. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 469–476, 2006.
- [12] 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.
- [13] T. Féraud, R. Chapuis, R. Aufrère, and P. Checchin. Improving Results of Non-Linear Observation Function Using a Kalman Filter Correction. In *International Conference on Information Fusion*, 2011. To appear.
- [14] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10, 2007.
- [15] N. M. Kwok and G. Dissanayake. An Efficient Multiple Hypothesis Filter for Bearing-Only SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 736–741, 2004.
- [16] J. Montiel, J. Civera, and A. J. Davison. Unified Inverse Depth Parametrization for Monocular SLAM. In *Robotics: Science and Systems*, 2006.
- [17] M. P. Parsley and S. J. Julier. Avoiding Negative Depth in Inverse Depth Bearing-Only SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2066–2071, 2008.
- [18] A. Petrovskaya and S. Thrun. Model Based Vehicle Detection and Tracking for Autonomous Urban Driving. *Autonomous Robots*, 26(2):123–139, 2009.
- [19] J. Shi and C. Tomasi. Good Features to Track. In *Computer Vision and Pattern Recognition*, pages 593–600, 1994.
- [20] R. Smith, M. Self, and P. Cheeseman. A Stochastic Map for Uncertain Spatial Relationships. In *4th International Symposium on Robotics Research*, pages 467–474, 1988.
- [21] J. Solà, A. Monin, M. Devy, and T. Lemaire. Undelayed Initialization in Bearing Only SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2499–2504, 2005.
- [22] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. FastSLAM: An Efficient Solution to the Simultaneous Localization And Mapping Problem with Unknown Data Association. *Journal of Machine Learning Research*, 4(3):380–407, 2004.