Hindawi
Journal of Sensors
Volume 2017, Article ID 7342931, 14 pages
https://doi.org/10.1155/2017/7342931

Research Article

Method for SLAM Based on Omnidirectional Vision: A Delayed-EKF Approach

Rodrigo Munguía, Carlos López-Franco, Emmanuel Nuño, and Adriana López-Franco

Department of Computer Science, CUCEI, University of Guadalajara, Guadalajara, Mexico

Correspondence should be addressed to Rodrigo Munguía; rodrigo.munguia@upc.edu

Received 4 October 2016; Accepted 4 January 2017; Published 19 February 2017

Academic Editor: Luis Payá

This work presents a method for implementing a visual-based simultaneous localization and mapping (SLAM) system using omnidirectional vision data, with application to autonomous mobile robots. In SLAM, a mobile robot operates in an unknown environment using only on-board sensors to simultaneously build a map of its surroundings, which it uses to track its position. The SLAM is perhaps one of the most fundamental problems to solve in robotics to build mobile robots truly autonomous. The visual sensor used in this work is an omnidirectional vision sensor; this sensor provides a wide field of view which is advantageous in a mobile robot in an autonomous navigation task. Since the visual sensor used in this work is monocular, a method to recover the depth of the features is required. To estimate the unknown depth we propose a novel stochastic triangulation technique. The system proposed in this work can be applied to indoor or cluttered environments for performing visual-based navigation when GPS signal is not available. Experiments with synthetic and real data are presented in order to validate the proposal.

1. Introduction

In recent years, many researchers have addressed the issue of making mobile robots more and more autonomous. In this context, the estimation of the state of the vehicle (i.e., its attitude and position) is a fundamental necessity for any application involving autonomy.

For open outdoor domains this problem is seemingly solved with on-board Global Positioning System (GPS) and Inertial Measurements Units (IMU), as well as their integrated version: the Inertial Navigation Systems (INS). On the contrary, unknown, cluttered, and GPS-denied environments still pose a considerable challenge. While attitude estimation is well handled by available systems, GPS-based position estimation can have some drawbacks. Specially GPS is not a reliable service as its availability can be limited by urban canyons and it is completely unavailable in indoor environments.

There are many approaches that work on this problem, one of the simplest is the use of odometry to estimate the pose of the robot; however, due to errors of the sensor and problems like wheel slip, this approach is useful only for short trajectories. For this reason, different type of sensors might be used. One option is the use of a laser sensor; this is an interesting approach; however these type of sensors are more expensive. Another option is the use of a camera; one of the advantages is the cost of this sensor which is very low compared with other sensors; in addition, its weight and power consumption make it an excellent option for robots with weight constraints, like an aerial robot; furthermore it provides more information than other sensors.

Simultaneous localization and mapping (SLAM) is an important problem to solve in robotics in order to build truly autonomous mobile robots. SLAM deals with the way in which a mobile robot can operate in an a priori unknown environment using only on-board sensors to simultaneously build a map of its surroundings, which it uses to track its position. Using visual-based SLAM methods, a robot can localize itself by detecting and tracking natural visual features of the environment. Also, because these visual features typically have an spatial meaning, the information obtained from the visual measurements can be used for replacing range

sensors: the lasers are often expensive and heavy, and the operation range of ultrasonics is limited.

On the other hand, while range sensors (e.g., laser) provide range and angular information, a camera is a projective sensor, which measures the bearing of image features. Therefore, depth information (range) cannot be obtained in a single frame [1, 2]. This fact has motivated the appearance of special techniques for feature initialization to allow the use of bearing sensors, such as cameras, in SLAM systems. In this sense, the treatment of the features in the stochastic map, such as, initialization and measurement., is perhaps still the most important problem in monocular SLAM in order to improve robustness.

Currently, there are two main approaches for implementing vision-based SLAM systems: (i) filtering-based methods (as [3–5]) and (ii) the optimization-based methods (as [6, 7]). While the former approach is shown to give accurate results when the availability of computational power is enough, filtering-based SLAM methods might be still beneficial if limited processing power is available [8]. Moreover, filtering-based methods are better suited for incorporating, in a simple manner, additional sensors to the system. In this sense, most robotics applications make use of multiple sensor inputs.

1.1. Related Work. There are several options for a visual sensor; the most common is a perspective camera; however this sensor has a short field of view, which can be a problem in an autonomous navigation system. In this work, we propose to use an omnidirectional vision system, which is a combination of a mirror and a camera [9, 10]; the advantages of this sensor are its wide field of view; this capability can improve the autonomous navigation system.

In previous works, omnidirectional vision systems have been used for localization. In [11], the authors present an approach for self-localization using an omnidirectional vision system and a perspective camera; they combine the information of both cameras and with the use of epipolar geometry they estimate the location of the robot. However the localization is estimated with respect to the omnidirectional images acquired previously; in contrast in our approach the localization of the robot is estimated with respect to the features acquired on real-time. In this case, the image features extracted from the omnidirectional vision sensor will be tracked and used as input for a method to estimate simultaneously both the pose of the robot and a map of its surroundings.

In [12] a Particle Filter is used for the localization of a mobile robot equipped with omnidirectional vision; an approach is proposed that reduces the number of features generated by SIFT, as well as their extraction and matching time. Omnidirectional vision can also be applied for the estimation of attitude and heading; in [13] a visual compass proposed makes use of catadioptric images in order to calculate the relative rotation and heading of an unmanned aerial vehicle. In [14] an algorithm that builds topological maps is proposed; in this case, local features are extracted from images obtained in sequence and are used for both: to cluster the images into nodes and to detect links between nodes.

Compared with the number of SLAM approaches using projective cameras, the use of omnidirectional vision in SLAM has been less explored. An early approach is proposed in [15], in this work two omnidirectional vision sensors are used as a stereo system for performing EKF-based SLAM. In [16] a monocular SLAM system that does not make use of any motion model is proposed. The key aspect of the system is a pose estimation algorithm that uses information from the estimated map, as well as the epipolar constraint.

In [17] a SLAM with an omnidirectional sensor has been proposed; in contrast with our approach they do not estimate the depth. In [18] the previous approach is extended by considering a patch formulation for data association which is invariant to rotation and scale. In [19], the authors present an approach for SLAM with omnidirectional vision; they extract features from the environment and update the position of the robot and a map library by using the EKF algorithm. In [20] the SLAM estimation method is based on the FastSLAM approach and the Hungarian algorithm for hierarchical data association. In [21] a technique for the extraction and matching of vertical lines in omnidirectional images is proposed; also, the initialization of features is carried out with an unscented transformation.

1.2. Objectives and Contributions. This work presents a novel method for implementing a visual-based SLAM system. The proposed approach combines odometry data and visual information obtained from an omnidirectional camera for estimating the state of the mobile robot as well as the map of its environment. The system can be applied to indoor or cluttered environments for performing visual-based navigation when no GPS signal is available. The system is mainly intended to be applied to small autonomous land robots, and its design takes into account the limited resources commonly available in this kind of applications. As it will be shown later, the proposal exhibits an improvement in the computational cost when it is compared with related methods.

The approach proposed in this paper shares some similarities with [18]; both are EKF-based methods and both use salient points as visual features. However while the method proposed in [18] makes use of the undelayed inverse depth technique [5] for initializing new features into the map, the proposed work is based on the delayed initialization technique, which is proposed in [22], for the same purpose.

Additionally this work presents the following novelties.

(i) The stochastic technique of triangulation that is used [22] for estimating the depth of visual features is extended in order to accommodate for the particularities of an omnidirectional vision sensor. In this case, a spherical projection model is used instead of the typical projection model in perspective cameras.

(ii) Also, different from [22] and other works, in order to improve the modularity and scalability of the proposed method, the process for detecting and tracking visual features is fully decoupled from the main estimation process. In this case, a simple but effective scheme is proposed.

(iii) Different from [18, 22], which make use of the inverse depth parametrization. In this work the features of the map are parametrized directly in Euclidean coordinates. Euclidean features imply less computational cost than the inverse depth features [23].

The paper is organized as follows: in Section 2 the description of the problem is presented, Section 3 presents an introduction to the projection model of the omnidirectional vision sensor, the proposed approach is presented in Section 4, in Section 5 the experiments results are presented, and finally in Section 6 the conclusions are given.

2. Problem Description

The main goal of the proposed method is to estimate the following system state x:

$$x = \left[x_v, y_1, y_2, \ldots, y_n\right]^T, \tag{1}$$

where x_v represents the state of the mobile robot and y_i represents the location of a feature point i in the environment. At the same time, x_v is composed of

$$x = \left[q^{NR}, \omega^R, r^N, v^N\right]^T, \tag{2}$$

where $q^{NR} = [q_1, q_2, q_3, q_4]$ represents the orientation of the vehicle with respect to the world (navigation) frame by a unit quaternion. $r^N = [p_x, p_y, p_z]$ represents the position of the vehicle (robot) expressed in the navigation frame. $\omega^R = [\omega_x, \omega_y, \omega_z]$ is the velocity rotation of the robot expressed in the same frame of reference. $v^N = [v_x, v_y, v_z]$ denote the linear velocity of the robot expressed in the navigation frame. The location of a feature y_i is parametrized in its Euclidean form:

$$y_i = \left[x_i, y_i, z_i\right]^T. \tag{3}$$

The proposed approach is intended for local autonomous vehicle navigation. In this case, the local tangent frame is used as the navigation reference frame and thus initial position of the vehicle defines its origin. The navigation coordinate frame follows the NED (North, East, Down) convention. In this work the magnitudes expressed in the (i) navigation frame, (ii) vehicle (robot) frame, and (iii) camera frame are denoted, respectively, by the superscripts N, R, and C (see Figure 1). All the coordinates systems are right-handed defined.

In this work, it is assumed that origin of the robot's coordinate frame R is aligned with the origin of the camera's coordinate C frame but rotated by a magnitude defined by a known camera-to-robot rotation matrix R^{CR}. A superscript AB denotes a reference frame B expressed with respect to reference A.

3. Omnidirectional Vision

As we mention previously, perspective cameras have a small field of view; one effective way to increase the field of view is the combination of a mirror with a camera; this approach is called omnidirectional vision. The image formation process of an omnidirectional vision sensor is different from the perspective camera. In this work, the model for omnidirectional cameras described in [24] is used. With this model we can compute the projection of a 3D point into the image plane, i.e., a projection $\mathbb{R}^3 \Rightarrow \mathbb{R}^2$.

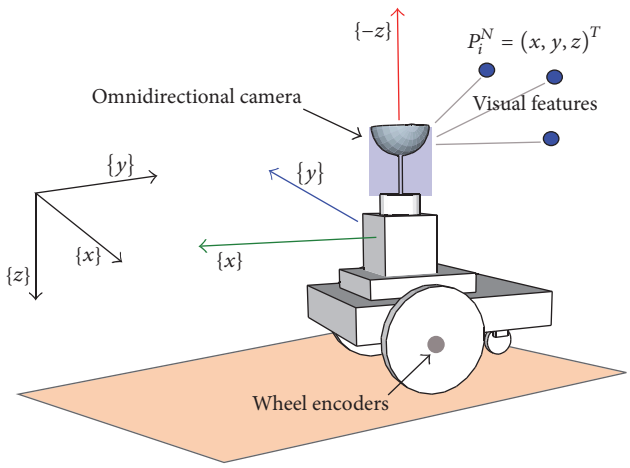

Figure 1: System parametrization. The local tangent coordinate frame is used as the navigation reference frame. It is assumed that origin of the robot's coordinate frame is aligned with the origin of the camera's coordinate frame but rotated by a known magnitude.

Let $P^N = (x, y, z)^T$ represent a 3D point; this point can be expressed in the camera frame by P^C; the relationship between these points can be defined as

$$P^C = R^{NC}\left(P^N - r^N\right), \tag{4}$$

where $R^{NC} = (R^{RN}R^{CR})^T$ is the rotation matrix transforming from navigation frame N to the camera frame C.

Then, the point P^C is projected onto a unit sphere (See Figure 2):

$$\chi_s = (x_s, y_s, z_s) = \frac{P^C}{\left\|P^C\right\|}. \tag{5}$$

The point χ_s is changed to a new reference frame centered in $C_p = (0, 0, \xi)$ and $\chi_{s'} = (x_s, y_s, z_s + \xi)$.

ξ is a calibration parameter of the model and depends only on the system modelled and the geometry of the mirror. For a common perspective camera $\xi = 0$. For catadioptric systems with parabolic mirror and orthographic camera $\xi = 1$. For systems with hyperbolic mirror and perspective camera $0 < \xi < 1$.

The point is then projected onto the normalized plane:

$$m_u = h\left(\chi_{s'}\right) = \left(\frac{x_{s'}}{z_{s'}}, \frac{y_{s'}}{z_{s'}}, 1\right). \tag{6}$$

Radial and tangential distortion induced by the lens are added through a distortion model D and distortion parameters $V = [k_1 \;\; k_2 \;\; k_3 \;\; k_4 \;\; k_5]$:

$$m_d = m_u + D\left(m_u, V\right). \tag{7}$$

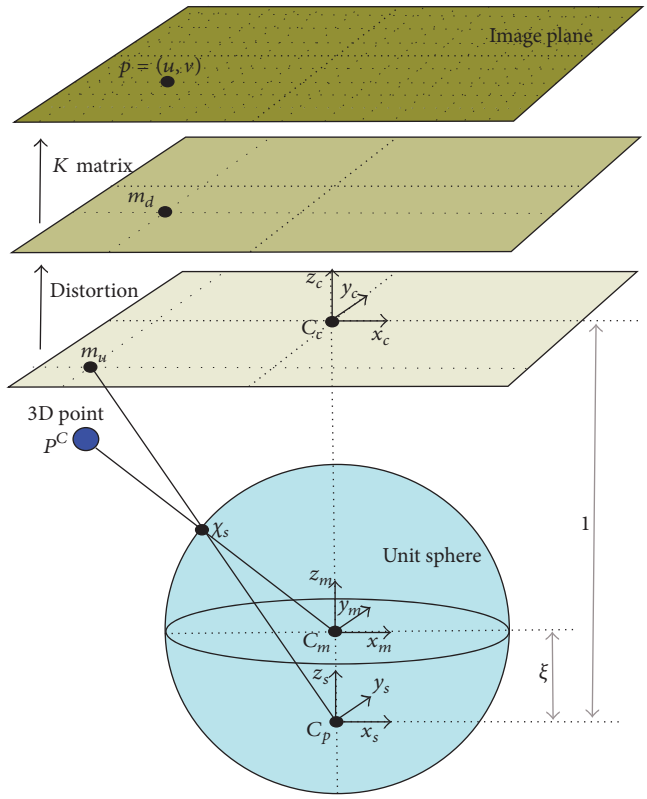

Figure 2: Full projection model used in this work.

The final projection involves a generalized camera projection matrix, where $\varsigma = \eta f$ is the generalized focal length for the camera-mirror system, η is a mirror parameter, f is the focal length of the camera, (u_0, v_0) is the principal point, s is the skew, and r is the aspect ratio.

$$p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \varsigma & \varsigma s & u_0 \\ 0 & \varsigma r & v_0 \\ 0 & 0 & 1 \end{bmatrix} m_d, \tag{8}$$

where u and v are the coordinates of the image point p expressed in pixels units.

Inversely, a directional vector $h^C = [h_x^C, h_y^C, h_z^C]^T$ can be computed from the image point coordinates u and v. The vector h^C points from the camera optical center position to the 3D point location.

$$h^C(m_u) = \begin{bmatrix} \dfrac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} x \\ \dfrac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} y \\ \dfrac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} - \xi \end{bmatrix}. \tag{9}$$

In (9), $(x, y, z) = m_u$, where m_u can be obtained from m_d from the inverted distortion model. And $m_d = K^{-1}[u, v, 1]^T$.

Finally, h^C can be expressed in the navigation frame by $h^N = R^{CN} h^C$ where $R^{CN} = R^{RN} R^{CR}$ is the camera to navigation rotation matrix.

4. Method Description

4.1. Detection and Tracking of Visual Measurements. As we mention before, in this work, an omnidirectional camera is considered (see Figure 1) as the main sensory input to the system. A standard small base-line tracker is proposed for detecting and tracking visual features. For this purpose, the Kanade-Lucas-Tomasi Tracker (KLT) [25] is used, but any small base-line tracker can be used too.

Detection of new features and their tracking are completely conducted by the KLT, (see Figure 3). In this case, the availability is assumed, at each k frame, of a list of $m_{(k)} = [z_{uv(1)}, z_{uv(2)}, \ldots, z_{uv(n)}]$ visual measurements of n static landmarks of the environment. Visual measurements z_{uv} are modelled by

$$z_{uv(i)} = y_{uv(i)} + v_{uv}, \tag{10}$$

where $y_{uv(i)} = [u, v]$ represents the true position in pixel coordinates (in the image plane) of the projection of a landmark i. The term v_{uv} represents the uncertainty associated with visual measurements and it is modelled by a Gaussian white noise with power spectral density (PSD) σ_{uv}^2.

Note that all the process for detecting and tracking visual features is fully decoupled from the main estimation process.

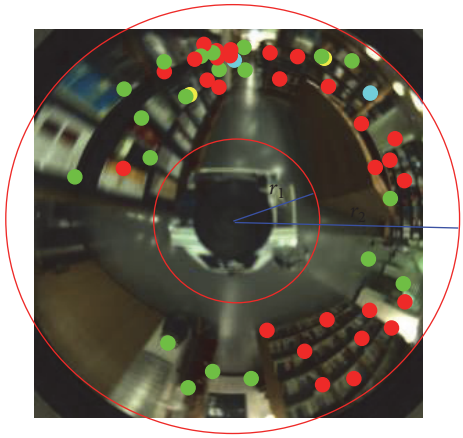

Figure 3: Detection and tracking of visual features are conducted by a Lucas Kanade tracker. Note that only the region of the image defined by $\{r_1 < r < r_2\}$ is considered, where r_1 is chosen in order to avoid visual features generated by the robot itself.

Later, it will be seen that the whole uncertainty associated with the tracking process will enter into the filter by means of the update equations through parameter σ_{uv}^2.

Most of the time, the KLT produces good quality results; however, sometimes, false positives are also obtained (e.g., continuous tracking of a feature that should be occluded by objects in front of it). In this work, a simple validation technique is used, which has been shown to give good results in combination with the KLT. This technique may be resumed as follows.

(i) When a new visual feature i is detected at frame k, then a $q \times q$ pixel window centered in $y_{uv(i)}$ is extracted. The patch is rotated n degrees by a bilinear transformation. Then, a smaller $p \times p$ pixel window $\text{pw}_{(i_j)}$ is extracted from the $q \times q$ pixel window, stored, and related to the i feature. The above process is repeated until a collection of j pixel windows $\text{pw}_{(i_j)}$ is obtained, representing the full rotation of the original $q \times q$ patch.

(ii) At every subsequent frame, $(k+1, k+2, \ldots, k+n)$, a new $p \times p$ pixel window, $\text{pw}_{(k+n)}$, centered in the current $y_{uv(i)}$ position found by the KLT is extracted. A patch cross-correlation technique is applied between the current pixel window, $\text{pw}_{(k+n)}$, and all the previously stored pixel windows, $\text{pw}_{(i_j)}$. If the score obtained is higher than a threshold, then the current (u_{d_i}, v_{d_i}) position is assumed to be a valid measurement.

The proposed technique rejects some overoptimistic measurements provided by the tracker, while it still provides some good degree of rotational invariance to rotations of the camera. The use of visual descriptors with at least some degree of invariance to rotations is very important in applications involving omnidirectional cameras.

Although some degree of validation is provided by the above technique, more robustness for the data association process can be obtained by the use of batch validation techniques as [26–28]. These techniques are applied during the update stage of the filter.

4.2. Odometry Measurements. In this work a differential wheeled robot is considered; however, it should be straightforward to adapt the proposed method to a platform with a different configuration.

Measurements $y_e = [u_r, u_l]^t$ of the rotational speed of the robot's wheels are obtained through the wheel encoders and can be modelled by

$$y_e = [\omega_r, \omega_l]^T + v_u, \tag{11}$$

where ω_r and ω_l represent the true angular rate of the right and left wheel and v_u is a Gaussian white noise with PSD σ_u^2.

4.3. System Prediction. At every step k, when measurements y_e of the encoders are available, the estimated system state $\hat{x}$ is taken a step forward by the following (discrete) nonlinear model:

$$\begin{aligned} q_{k+1}^{NR} &= \left(\cos\|w\| I_{4\times4} + \frac{\sin\|w\|}{\|w\|} W\right) q_k^{NR}, \\ \omega_{k+1}^{R} &= \left[0, \omega_{y_k}, \omega_z\right]^T, \\ r_{k+1}^{N} &= r_k^N + v_k^N \Delta t, \\ v_{k+1}^{N} &= R^{RN}\left[v_x, 0, 0\right]^T, \end{aligned} \tag{12}$$

where R^{RN} is computed from the current quaternion q_k^{NR}, Δt is the sample time of the system, and

$$\begin{aligned} v_x &= \frac{r}{2\left(u_r + u_l\right)}, \\ \omega_z &= \frac{r}{L\left(u_r - u_l\right)}, \end{aligned} \tag{13}$$

where r is the wheel radius, L is the distance between wheels of the robot, and u_r, u_l are the angular rotation measured by the encoders of right and left wheels. In the model defined in (12), a closed form solution of $\dot{q} = 1/2(W)q$ is used for integrating the current velocity rotation ω^R over the quaternion q^{NR}. In this case $w = [\omega_{k+1}^R \Delta t/2]^\top$ and

$$W = \begin{bmatrix} 0 & -w_1 & -w_2 & -w_3 \\ w_1 & 0 & -w_3 & w_2 \\ w_2 & w_3 & 0 & -w_1 \\ w_3 & -w_2 & w_1 & 0 \end{bmatrix}. \tag{14}$$

In the system prediction model presented above, note that a typical differential drive kinematics model, (13), which is defined for flat surfaces, has been coupled with a more general 6DOF kinematic model defined in (12). This modular design should allow using another kind of motion models with minor modifications.

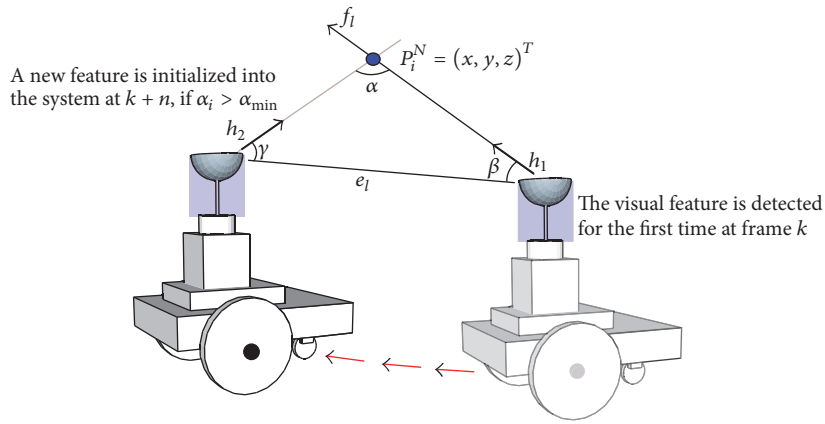

Figure 4: A hypothesis of depth is computed for each visual feature prior to its inclusion into the system state by means of a stochastic technique of triangulation.

An Extended Kalman Filter (EKF) propagates the system state $\hat{x}$ over time. It is assumed that the map features $\hat{y}_i$ remain static (rigid scene assumption) so $\hat{x}_{k+1} = [\hat{x}_{v(k+1)}, \hat{y}_{1(k)}, \hat{y}_{2(k)}, \ldots, \hat{y}_{n(k)}]^\top$. The state covariance matrix P is taken a step forward by

$$P_{k+1} = \nabla F_x P_k \nabla F_x^T + \nabla F_u Q \nabla F_u^T, \tag{15}$$

where Q and the Jacobians ∇F_x, ∇F_u are defined as follows:

$$\nabla F_x = \begin{bmatrix} \dfrac{\partial f_v}{\partial \hat{x}_v} & 0_{13\times n} \\ 0_{n\times 13} & I_{n\times n} \end{bmatrix},$$

$$\nabla F_u = \begin{bmatrix} \dfrac{\partial f_v}{\partial u} & 0_{13\times n} \\ 0_{n\times 2} & 0_{n\times n} \end{bmatrix}, \tag{16}$$

$$Q = \begin{bmatrix} U & 0_{2\times n} \\ 0_{n\times 2} & 0_{n\times n} \end{bmatrix}.$$

$\partial f_v/\partial \hat{x}_v$ are the derivatives of the equations of the nonlinear prediction model (12) with respect to the robot state $\hat{x}_v$. $\partial f_v/\partial u$ are the derivatives of the nonlinear prediction model with respect to the system inputs y_g and y_a. Uncertainties of the encoders, as well another kind of unstructured uncertainties, like wheel slippages, are incorporated into the system by means of the process noise covariance matrix $U = \sigma_u^2 I_{2\times 2}$, through parameter σ_u^2.

4.4. Visual Aid. Depth information cannot be obtained in a single measurement when bearing sensors (e.g., a omnidirectional camera) are used. To infer the depth of a feature, the sensor must observe this feature repeatedly as it freely moves through its environment, estimating the angle from the feature to the sensor center. The difference between those angle measurements is the parallax angle. Actually, parallax is the key to estimate the depth of the features. In case of indoor sequences, a displacement of centimeters could be enough to produce parallax; on the other hand, when the distance to a feature increases, then the sensor has to travel more to produce parallax.

In monocular-based systems, the treatment of the features in the stochastic map (initialization, measurement, etc.) is an important problem to address with direct implications in the robustness of the system. In this work, a novel method is proposed, for incorporating new features into the system. In this approach, a single hypothesis is computed for the initial depth of features, by using of a stochastic technique of triangulation. The method is based on previous author's work [22]. In this previous work, a common projective camera is used as input to the SLAM system.

4.4.1. Initialization of Visual Features. At the k frame, when a visual feature is detected for the first time, the following entry f_l is stored in a table (see Figure 4):

$$f_l = \left[t_{c_0}^N, \theta_0, \phi_0, P_{y_i}\right], \tag{17}$$

where $y_i = [t_c^N, \theta, \phi]$ models a 3D semiline defined on one side by the vertex t_c^N, corresponding to the current optical center coordinates of the camera expressed in the navigation frame and pointing to infinity on the other side, with azimuth and elevation, θ and ϕ, respectively, and

$$\theta = \operatorname{atan2}\left(h_y^N, h_x^N\right),$$

$$\phi = \operatorname{acos}\left(\frac{h_z^N}{\sqrt{\left(h_x^N\right)^2 + \left(h_y^N\right)^2 + \left(h_z^N\right)^2}}\right), \tag{18}$$

where $h^N = [h_x^N, h_y^N, h_z^N]^T$ is computed as is indicated in Section 3. P_{y_i} is a 5×5 covariance matrix which models the uncertainty of y_i. $P_{y_i} = JPJ^T$, where P is the system covariance matrix and J is the Jacobian matrix formed by the partial derivatives of the function $y_i = h(\hat{x}, z_{uv})$ with respect to $[\hat{x}, z_{uv}]^T$.

At every subsequent frame $k+1, k+2, \ldots, k+n$, a hypothesis of the feature depth d_i is computed by (see Figure 4)

$$d_i = \frac{\|e_l\| \sin\gamma}{\sin\alpha}, \tag{19}$$

where $\alpha_i = \pi - (\beta + \gamma)$ is the parallax. $e_l = t_{c_0}^N - t_c^N$ indicates the displacement of the camera since it was first observed to its current position, and

$$\begin{aligned} \beta &= \cos^{-1}\left(\frac{h_1 \cdot e_l}{\|h_1\| \|e_l\|}\right), \\ \gamma &= \cos^{-1}\left(\frac{h_2 \cdot -e_l}{\|h_2\| \|e_l\|}\right), \end{aligned} \tag{20}$$

where β is the angle defined by h_1 and e_l. h_1 is the normalized directional vector m:

$$m(\theta_i, \phi_i) = (\cos\theta \sin\phi, \sin\theta \sin\phi, \cos\phi)^T \tag{21}$$

computed taking θ_i, ϕ_i from f_l and where γ is the angle defined by h_2 and $-e_l$. $h_2 = h^N$ is the directional vector pointing from the current camera optical center to the feature location and is computed as is indicated in Section 3 from the current measurement z_{uv}.

At each step, the hypothesis of depth d_i is passed through a low-pass filter. The above occurs since the depth estimated by triangulation varies considerably, especially for low parallax. In previous author's work [22] it is showed that only a few degrees of parallax are enough in order to reduce the uncertainty in depth estimations.

When the parallax α_i is greater than a threshold ($\alpha_i > \alpha_{\min}$) a new feature $\hat{y}_{\text{new}} = [x_i, y_i, z_i]^\top$ is added to the system state vector $\hat{x}$:

$$\hat{x}_{\text{new}} = [\hat{x}_{\text{old}}; \hat{y}_{\text{new}}]^\top, \tag{22}$$

where

$$\hat{y}_{\text{new}} = t_{c_0}^N + m(\theta_i, \phi_i)\, d. \tag{23}$$

In this work, good experimental results were obtained with $\alpha_{\min} = 4\,\text{deg}$.

The system state covariance matrix $\hat{P}$ is updated by

$$\hat{P}_{\text{new}} = \begin{bmatrix} \hat{P}_{\text{old}} & 0 \\ 0 & P_{y_{\text{new}}} \end{bmatrix}, \tag{24}$$

where $P_{y_{\text{new}}}$ is the 3×3 covariance matrix which models the uncertainty of the new feature $\hat{y}_{\text{new}}$, and

$$P_{y_{\text{new}}} = J \begin{bmatrix} P_{y_i} & 0 \\ 0 & \sigma_d^2 \end{bmatrix} J^T. \tag{25}$$

In (25), P_{y_i} is taken from f_l and σ_d^2 is a parameter modelling the uncertainty of process of depth estimation. J is the Jacobian matrix formed by the partial derivatives of the function $\hat{y}_{\text{new}} = h(f_l, d)$ with respect to $[t_{c_0}^N, \theta_0, \phi_0, d]^T$.

4.4.2. Visual Updates. Assuming that, for the current frame, n visual measurements $z_{uv_1}, z_{uv_2}, \ldots, z_{uv_n}$ are available for n features $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ (see Section 4.1), then the filter is updated with the Kalman update equations as follows:

$$\begin{aligned} \hat{x}_k &= \hat{x}_{k+1} + K(z - h), \\ P_k &= P_{k+1} - KSK^T, \\ K &= P_{k+1} \nabla H^T S^{-1}, \\ S_i &= \nabla H P_{k+1} \nabla H^T + R, \end{aligned} \tag{26}$$

where $z = [z_{uv_1}, z_{uv_2}, \ldots, z_{uv_n}]^T$ is the current measurements. $h = [h_1, h_2, \ldots, h_n]^T$ is the current prediction measurements. The measurement prediction model $h_i = (u, v) = h(\hat{x}_v, \hat{y}_i)$ is defined in Section 3. K is the Kalman gain. S is the innovation covariance matrix. $\nabla H = [\nabla H_1, \nabla H_2, \ldots, \nabla H_n]^T$ is the Jacobian formed by the partial derivatives of the measurement prediction model $h(\hat{x})$ with respect to the state $\hat{x}$.

$$\nabla H_i = \left[\frac{\partial h_i}{\partial \hat{x}_v}, \ldots, 0_{2\times 3}, \ldots, \frac{\partial h_i}{\partial \hat{y}_i}, \ldots, 0_{2\times 3}, \ldots\right], \tag{27}$$

where $\partial h_i / \partial \hat{x}_v$ are the partial derivatives of the equations of the measurement prediction model h_i with respect to the robot state $\hat{x}_v$. And $\partial h_i / \partial \hat{y}_i$ are the partial derivatives of h_i with respect to feature $\hat{y}_i$. Note that $\partial h_i / \partial \hat{y}_i$ have only a nonzero value at the location (indexes) of the observed feature $\hat{y}_i$. $R_i = (I_{2n\times 2n})\sigma_{uv}^2$ is the measurement noise covariance matrix.

4.4.3. Map Management and Real-Time Issues. A SLAM framework that works reliably locally can be applied to large-scale problems using methods, such as submapping, graph-based global optimization [8], or global mapping.

Therefore, in this work, large-scale SLAM and loop-closing are not considered. Although, these problems have been intensively studied in the past.

Moreover, this work is motivated by the application of monocular SLAM in the context of visual odometry. When the number of features in the system state increases, then computational cost grows rapidly, and consequently, it becomes difficult to maintain the frame rate operation. To alleviate this drawback, old features can be removed from the state for maintaining a stable number of features and, therefore, to stabilize the computational cost per frame. Obviously, if old features are removed, then previous mapped areas cannot be recognized in the future. However, in the context of visual odometry or local mapping, this fact is not considered as a problem.

In particular, in this work, oldest features are removed from the system state when the number of map features surpasses a certain threshold. Good experimental results were obtained with a threshold equal to 50 features. The removal process is carried out using the approach described in [22].

It is important to note that, since early works as [29], the real-time feasibility of EKF-based SLAM methods was

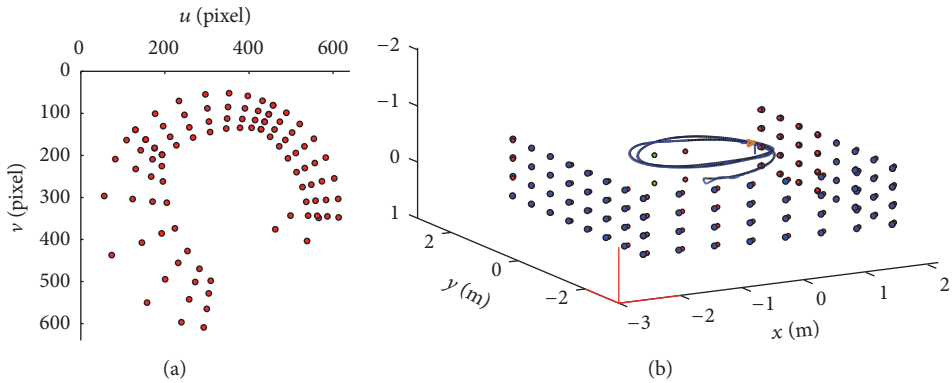

Figure 5: In simulations, the omnidirectional camera was moved through an environment composed by 3d points which emulate visual landmarks. (a) shows one of the frames as it should have been seen by the camera. (b) shows the actual (red) and estimated (blue) trajectory and map after 500 frames.

shown for maps composed up to 100 features using standard hardware. Even though there are real-time implementations for high computation demanding techniques as the optimization-based methods, the threshold used in this work is under a bound that should permit a real-time performance, by means of an optimized implementation.

As it was mentioned before, in the proposed method, the process for detecting and tracking visual features is fully decoupled from the main estimation process. In this sense, this architecture should permit to parallelize at least both processes in software or hardware implementations. However, this subject is beyond the scope of this work.

5. Experimental Results

In this section we present the results obtained using synthetic data obtained by simulations as well as the results obtained with real data. The experiments were performed in order to validate the performance of the proposed method. A MATLAB® implementation was used for this purpose.

5.1. Experiments with Simulations. The main purpose of the experiments, carried out through simulations, was to validate the correctness of the measurement model used for the omnidirectional camera. In this case, the problem of data association was avoided and perfect matching of visual features was assumed for experiments. Also in simulations, the use of the odometry data obtained from the robot's encoders was avoided. In this case, the model described in (12) was slightly modified in order to use Gaussian noise as inputs instead of odometry measurements. With this modification, the method was tested in a purely monocular 6DOF SLAM context, for obtaining more insight about the performance of the visual aiding.

Figure 5 shows the simulated environments used in experiments. In this case, the camera was moved in order to follow a spiral-like trajectory. The area, where the vehicle is moving, is surrounded by a wall of landmarks. During the trajectory, the camera yaw was changed under a constant pattern of movement. In order to recover the metric scale of the world, the system was initialized with perfect knowledge of four landmarks. At the end of the experiment it can be appreciated that both trajectory and map have been satisfactorily estimated.

5.2. Comparative Study with Real Data. The proposed method was executed off-line, using the Rawseeds dataset [30] as input signals, in order to test its performance with real data. This dataset is freely available online, and it contains several sources of sensors signals. In the experiments the following signals were used: (i) visual information obtained from an omnidirectional camera at 15 frames per second (fps) and with a resolution of 640 × 640 pixels. (ii) Odometry data was obtained from the wheels encoders of the robot, available at 50 Hz. The sensors are mounted over a differential drive platform, (see [30] for a complete description). In experiments, the 1-point RANSAC method [28] has been used for validating the visual matches of map features. The method was tested under both indoor and outdoor conditions. Figure 6 shows examples of the visual input, as well as the output of the proposed method for both cases.

For comparison purposes, another two related methods were also implemented: (i) the scheme proposed in [18] (UID), which is based in the well-known undelayed inverse depth methodology and (ii) A variant of the former approach, the undelayed inverse depth to Euclidean method (UID2E), which is based on [23]. For performing the experiments, the same input signals were used with the three methods. In this case, because the process for detecting and tracking visual features is fully decoupled from the main estimation process (Section 4.1), it is important to note that the same set of visual features were used as input to the three methods. Therefore, the experimental results were obtained under similar conditions.

5.2.1. Indoor Experiments. The dataset used for testing the proposed method under indoor conditions was captured by the robot traveling through corridors and hallways of a building intended to education. The sequence used in these experiments is composed by 3700 frames. Figure 7 shows

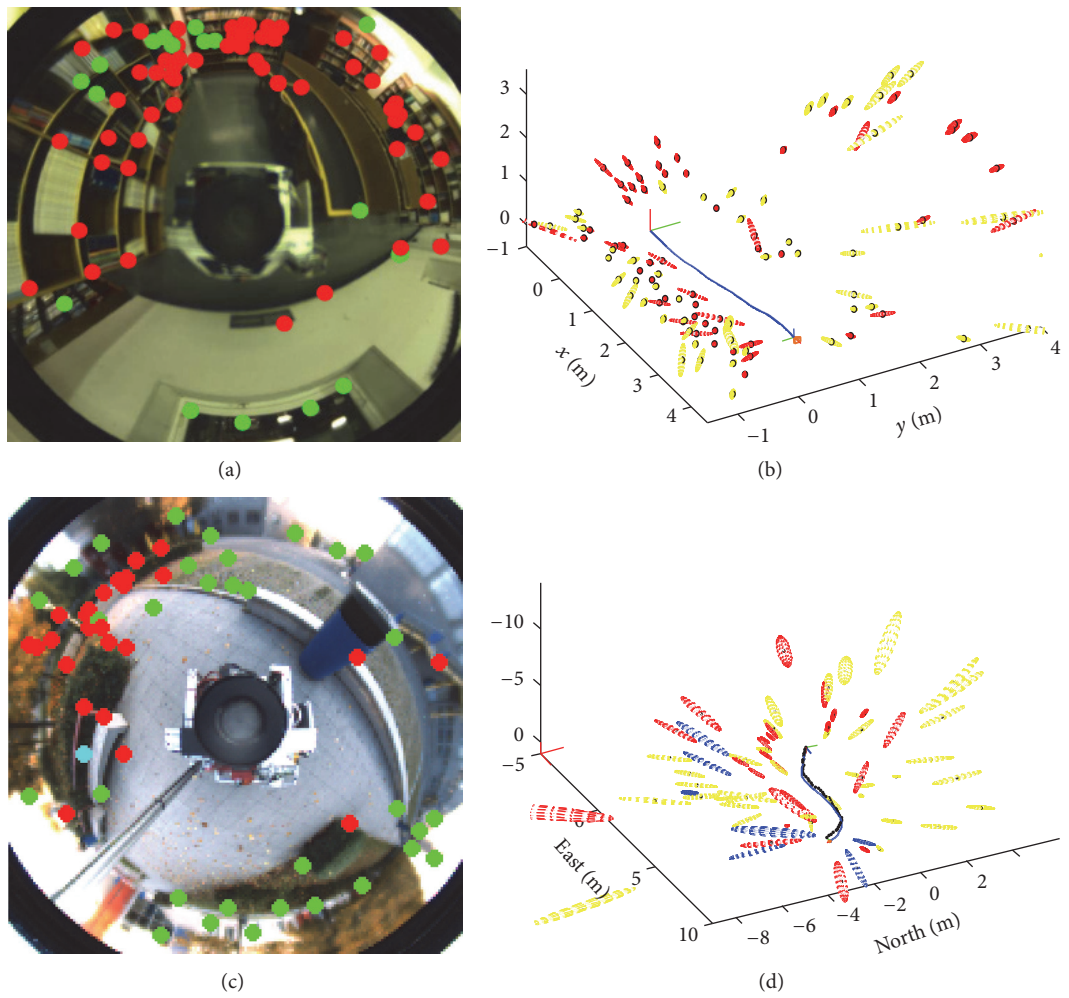

Figure 6: Examples of the experiments with real data in indoor (a, b) and outdoor (c, d) environments. The video frames (a, c) are displaying illustrating visual features that have been tracked, as well as the corresponding trajectory and map estimated by the method (b, d). Comparing visual features corresponding to elements of the environment with the estimated map, it can be appreciated that the physical structure of the environment is partially recovered.

Table 1: Experimental results obtained under indoor conditions.

Method	Ts (s)	aTpFs (s)	Tr (s)	aTpFr (s)	NIF	aFpF	aNFRpF
DE	130	$.03 \pm .01\sigma$	114	$.03 \pm .03\sigma$	1294	$35.4 \pm 9.7\sigma$	$1.0 \pm 2.2\sigma$
UID	210	$.06 \pm .03\sigma$	525	$.15 \pm .14\sigma$	2126	$59.0 \pm 13\sigma$	$2.1 \pm 3.8\sigma$
UID2E	201	$.06 \pm .03\sigma$	384	$.11 \pm .10\sigma$	2304	$59.3 \pm 13\sigma$	$2.6 \pm 4.2\sigma$

a top view of the map and trajectory that was obtained with each method. It is important to note that for indoor conditions there is no availability of a ground truth signal. In this case, in order to have a better interpretation of the results, the estimated map and trajectory were overlaid to a CAD drawing of the building where the raw data was collected.

Table 1 summarizes the experimental results, under indoor environments, obtained with the three methods. The following statistics have been computed for each method: (i) the execution time of the SLAM process (Ts), (ii) the average execution time per frame of the SLAM process (aTpFs), (iii) the execution time of the 1-point-RANSAC validation process (Tr), (iv) the average execution time per frame of the 1-point-RANSAC validation process (aTpFr), (v) the number of features initialized into the system (NIF), (vi) the average number of features per frame used for updating the filter (aFpF), and (vii) the average number of outliers rejected per frame, by the 1-point-RANSAC validation process (aNFRpF).

Figure 8 illustrates the evolution over time of the number of features contained within the system state. Figure 9 illustrates the evolution over time of the number of outliers detected by 1-point-RANSAC method. The visual matches detected as outliers are not considered for updating the filter.

FIGURE 7: Top view of the estimated map and trajectory, using real data from sensors mounted in a differential drive robot, moving through corridors and hallways of a building. The results are presented for the proposed method (DE) (a), the UID method (b), and the UID2E method (c).

5.2.2. Outdoor Experiments. The dataset used for testing the proposed method under outdoor conditions was captured by the robot traveling along sidewalks of a campus. The sequence used in these experiments is composed of 1600 frames. Figure 10 shows the map and trajectory obtained with the proposed method (DE) and the UID and UID2E methods. Note that for outdoor experiments a ground truth reference is available.

Table 2 summarizes the experimental results under outdoor environments obtained with the three methods. Additionally to the statistics computed for indoor experiments, in this case for outdoor experiments, the average Euclidean

Table 2: Experimental results obtained under outdoor conditions.

Method	Ts (s)	aTpFs (s)	Tr (s)	aTpFr (s)	NIF	aFpF	aNFRpF	AEE (m)
DE	148	$.09 \pm .04\sigma$	44	$.02 \pm .03\sigma$	755	$44.6 \pm 7.5\sigma$	$0.3 \pm 0.6\sigma$	$0.8 \pm 0.4\sigma$
UID	300	$.19 \pm .10\sigma$	236	$.15 \pm .10\sigma$	1442	$89.4 \pm 21\sigma$	$1.6 \pm 1.6\sigma$	$2.1 \pm 1.7\sigma$
UID2E	241	$.15 \pm .07\sigma$	142	$.09 \pm .06\sigma$	1399	$89.3 \pm 21\sigma$	$1.3 \pm 1.6\sigma$	$2.1 \pm 1.6\sigma$

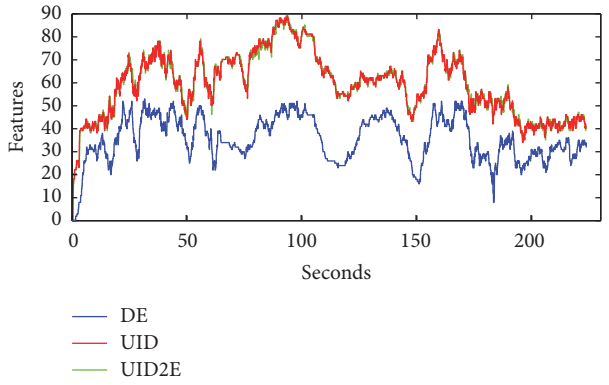

Figure 8: Indoor experiment: evolution of the number of the map features for each method.

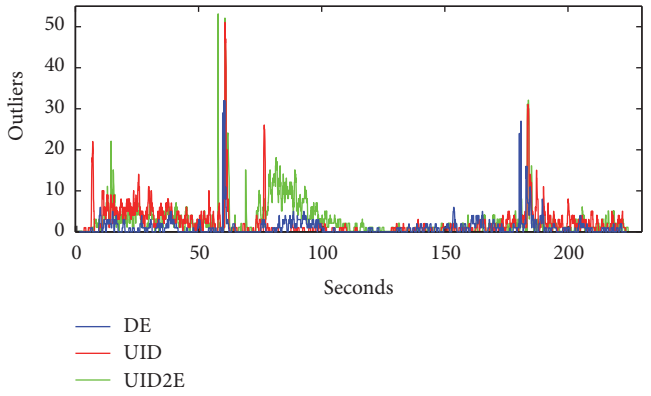

Figure 9: Indoor experiment: evolution of the number of outliers rejected per frame by the 1-point-RANSAC method.

error (AEE) in position estimation was also computed due to the availability of the ground truth signal.

Figure 11 shows the evolution over time of the Euclidean error in position estimation for each method. Note that this figure also displays the Euclidean error obtained when only odometry data, obtained from the wheels encoders of the robot, is used for computing the position. Figures 12 and 13, respectively, show the evolution over time of the number of features included within the map and the number of outliers rejected per frame by the 1-point-RANSAC method.

5.2.3. Discussion. According to the results of the comparative study some implications can be inferred.

For indoor experiments, analysing Figure 7, it can be seen that similar trajectories were estimated with the proposed method (DE) and the UID method. On the other hand, the UID2E method presents a considerable drift in this experiment. For outdoor experiments, by analysing Figures 10 and 11, it can be seen that the DE method showed a slightly better performance. In this case, very similar results were obtained with UID and UID2E methods.

After analyzing the results presented in Tables 1 and 2, it can be observed that the proposed method offers a better performance in terms of computational cost. Two main reasons can explain the reduction of the execution time of the DE method with respect to UID and UID2E methods.

The first reason has to do with the use of the Euclidean parametrization of features, which results in a reduction of the size of the vector state (compared with the inverse depth parametrization). As it is well known, the computational cost of the Kalman Filter scales badly with the size of the state. The UID2E method transforms the inverse depth features, that are well conditioned, to Euclidean features in order to improve the computational cost of the UID method. In fact, by comparing with the UID method, some reduction of computational cost is obtained with the UID2E method. However, as it was the case with indoor experiments, the UID2E method can show sometimes a larger drift in estimations.

The second reason has to do with the inherent property of the proposed method to reject weak features during the initialization process. Different from the undelayed methods whose features are initialized at the very first time that they are observed, the delayed methods (as the proposed one) collect depth information of a feature prior to its inclusion into the system state. During this initial period, weak candidate points that are detected but are lost after only a few frames that have been tracked are not initialized into the system state. Also, candidate points that do not exhibit parallax are not included into the system. As a consequence of the above, a less number of strong visual features are initialized into the system with the DE method (observe the NIF column in Tables 1 and 2).

On the other hand, as it was mentioned before, undelayed methods (UID and UID2E) initialize into the filter all the visual features detected. In this case, weak visual features are continuously been removed from the system state, while the new features are been detected and initialized. The high number of features that are initialized and removed continually requires that some portion of the system state performs somehow as a buffer for these features. This has as a consequence that the UID and the UID2E methods require maintaining a larger system state (see Figures 8 and 12).

The 1-point-RANSAC validates the unequivocal association of visual features with its spatial meaning. This validation

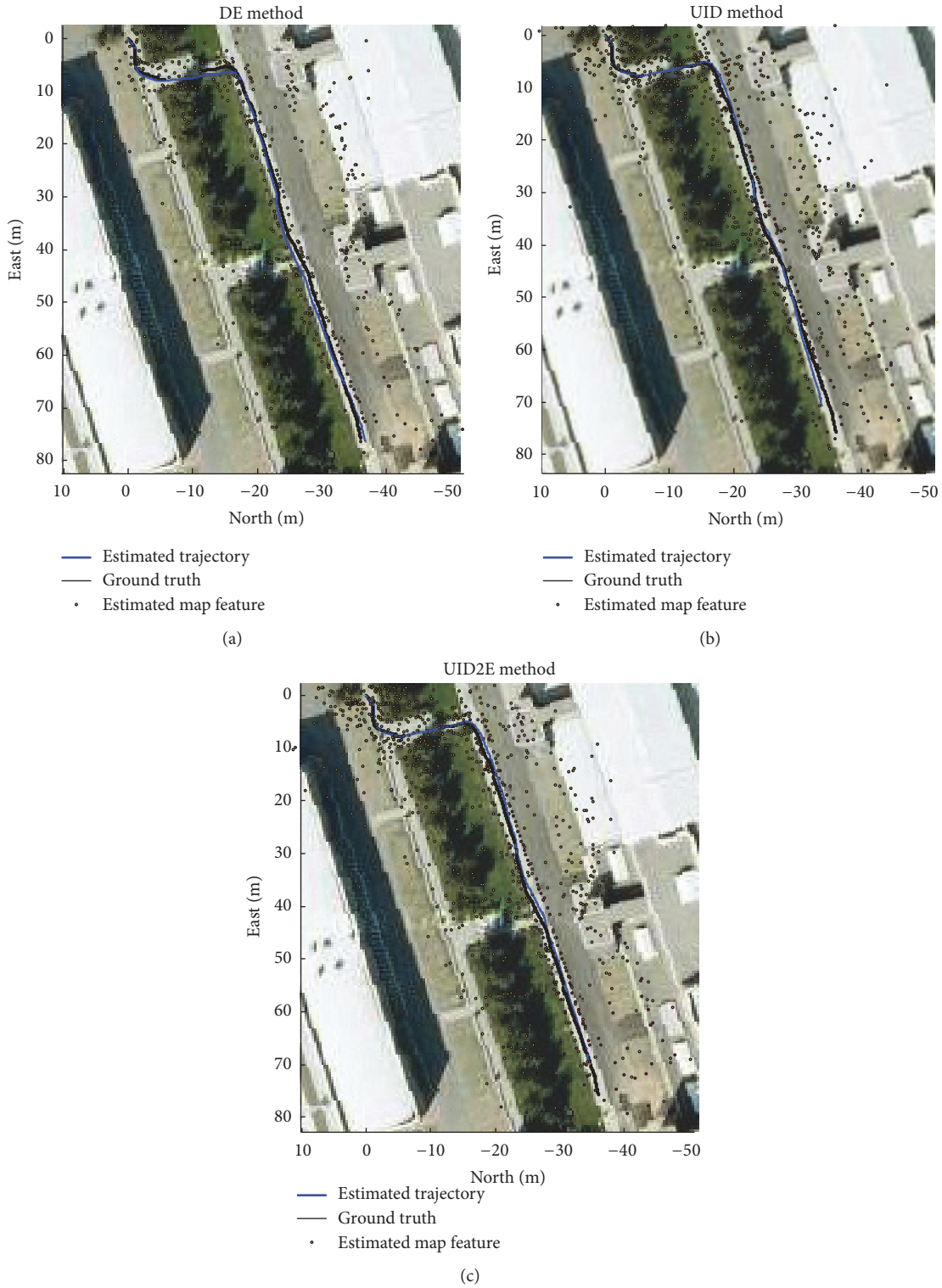

Figure 10: Aerial view of the estimated map and trajectory which were obtained in outdoor experiments for each method.

process rejects outliers by checking statistical consistency of visual measurements with the features map. As it was mentioned above, the same set of visual measurements were used as input to the three methods. Therefore, a lower percentage of outliers should imply a better statistical consistency in estimation. In this case, it is interesting to note that the proposed method (DE) also shows a lower percentage of outliers detected by the 1-point-RANSAC.

See the column aNFRpF in Tables 1 and 2 and also see Figures 9 and 11.

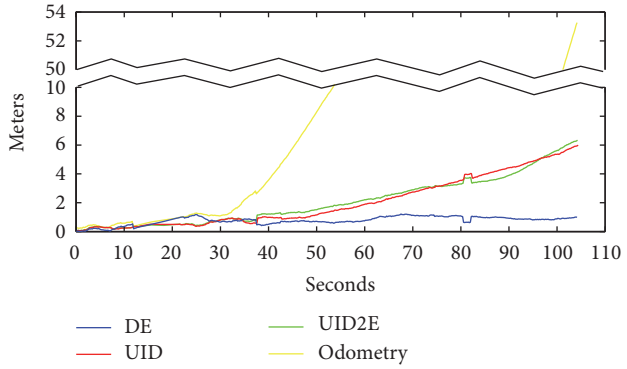

Figure 11: Outdoor experiment: evolution of the Euclidean error in position estimation.

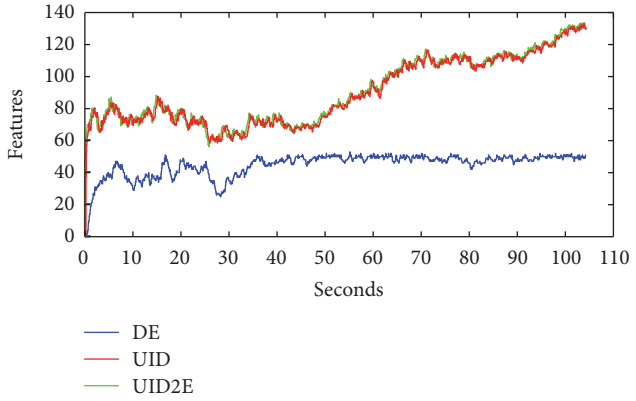

Figure 12: Outdoor experiment: evolution of the number of the map features for each method.

This suggests that the DE method produces maps less dense of features, but also more consistent.

Moreover, the number of outliers also has a huge impact to the computational cost. In this case, the time consumed by the 1-point-RANSAC validation process increases with the number of outliers (see Tables 1 and 2).

6. Conclusion

In this paper a method for mobile robot navigation using visual information is presented. The proposed scheme is a filter-based simultaneous localization and mapping (SLAM) system. In this case, visual information is incorporated into the system in order to minimize the odometry error. A monocular omnidirectional vision sensor was used in this work; this sensor allows tracking the features more time during the navigation and this makes the approach more robust to estimate the pose of the mobile robot as well as the map of the environment. In this case, the appearance and spatial information provided by the built map could be useful for multiple tasks. To estimate the depth of the features a novel method is proposed which is based in a stochastic technique of triangulation. In addition, the detection and tracking process are fully decoupled from the main estimation process. For this purpose, a simple but effective scheme was proposed.

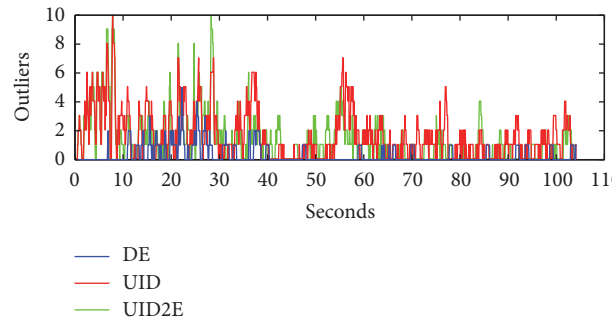

Figure 13: Outdoor experiment: evolution of the number of outliers rejected per frame by the 1-point-RANSAC method.

Finally, the effectiveness of the proposed approach is verified through simulations and experimental results with real data.

A comparison study was presented in order to provide additional insights about the performance of the proposed method. When it is compared with the related methods that were used in the study, the proposal shows a considerable improvement in terms of computational cost. Also, the results suggest that more consistent estimates are obtained with the proposed method. The proposed method should be useful to be applied to mobile robots moving across indoor or cluttered environments where GPS signal is not available.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] C. Löpez-Franco and E. Bayro-Corrochano, "Omnidirectional vision and invariant theory for robot navigation using conformal geometric algebra," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, pp. 570–573, IEEE, Hong Kong, August 2006.

[2] C. López-Franco and E. Bayro-Corrochano, "Omnidirectional vision for visual landmark identification using p 2-invariants," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, pp. 545–550, May 2006.

[3] R. Munguía, B. Castillo-Toledo, and A. Grau, "A robust approach for a filter-based monocular simultaneous localization and mapping (SLAM) system," *Sensors*, vol. 13, no. 7, pp. 8501–8522, 2013.

[4] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[5] J. M. M. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Proceedings of the 2nd International Conference on Robotics Science and Systems (RSS '06)*, pp. 81–88, August 2006.

[6] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular SLAM," in *Proceedings of the International Conference on Robotics Science and Systems (RSS '10)*, pp. 73–80, Zaragoza, Spain, June 2010.

[7] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '07)*, pp. 225–234, IEEE, Nara, Japan, November 2007.

[8] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-time monocular SLAM: why filter?" in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 2657–2664, IEEE, May 2010.

[9] R. A. Hicks and R. K. Perline, "Geometric distributions for catadioptric sensor design," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, pp. I584–I589, December 2001.

[10] C. López-Franco and E. Bayro-Corrochano, "Unified model for omnidirectional vision using the conformal geometric algebra framework," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 4, pp. 48–51, Cambridge, UK, August 2004.

[11] L. Puig and J. J. Guerrero, "Self-location from monocular uncalibrated vision using reference omniviews," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '09)*, pp. 5216–5221, St. Louis, Mo, USA, October 2009.

[12] H. Tamimi, H. Andreasson, A. Treptow, T. Duckett, and A. Zell, "Localization of mobile robots with omnidirectional vision using Particle Filter and iterative SIFT," *Robotics and Autonomous Systems*, vol. 54, no. 9, pp. 758–765, 2006, Selected papers from the 2nd European Conference on Mobile Robots (ECMR 05)2nd European Conference on Mobile Robots.

[13] I. F. Mondragón, P. Campoy, C. Martinez, and M. Olivares, "Omnidirectional vision applied to Unmanned Aerial Vehicles (UAVs) attitude and heading estimation," *Robotics and Autonomous Systems*, vol. 58, no. 6, pp. 809–819, 2010.

[14] C. Valgren, A. Lilienthal, and T. Duckett, "Incremental topological mapping using omnidirectional vision," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 3441–3447, Beijing, China, October 2006.

[15] J.-H. Kim and M. J. Chung, "SLAM with omni-directional stereo vision sensor," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 442–447, Las Vegas, Nev, USA, October 2003.

[16] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08)*, pp. 2531–2538, Nice, France, September 2008.

[17] A. Rituerto, L. Puig, and J. J. Guerrero, "Visual SLAM with an omnidirectional camera," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 348–351, IEEE, Istanbul, Turkey, August 2010.

[18] D. Gutierrez, A. Rituerto, J. M. M. Montiel, and J. J. Guerrero, "Adapting a real-time monocular visual SLAM from conventional to omnidirectional cameras," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops '11)*, pp. 343–350, November 2011.

[19] K. Wang, G. Xia, Q. Zhu, Y. Yu, Y. Wu, and Y. Wang, "The SLAM algorithm of mobile robot with omnidirectional vision based on EKF," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA '12)*, pp. 13–18, Beijing, China, June 2012.

[20] C. Gamallo, M. Mucientes, and C. V. Regueiro, "A FastSLAM-based algorithm for omnidirectional cameras," *Journal of Physical Agents*, vol. 7, no. 1, pp. 12–21, 2013.

[21] T. Phan and A. Ovchinnikov, "Bearing-only simultaneous localization and mapping using omnidirectional camera," in *Some Current Advanced Researches on Information and Computer Science in Vietnam*, Q. A. Dang, X. H. Nguyen, H. B. Le, V. H. Nguyen, and V. N. Q. Bao, Eds., vol. 341 of *Advances in Intelligent Systems and Computing*, pp. 107–121, Springer International Publishing, New York, NY, USA, 2015.

[22] R. Munguía and A. Grau, "Monocular SLAM for visual odometry: a full approach to the delayed inverse-depth feature initialization method," *Mathematical Problems in Engineering*, vol. 2012, Article ID 676385, 26 pages, 2012.

[23] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth to depth conversion for monocular SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 2778–2783, Roma, Italy, April 2007.

[24] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 3945–3950, IEEE, Roma, Italy, April 2007.

[25] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593–600, Seattle, Wash, USA, June 1994.

[26] T. Baile, *Mobile robot localisation and mapping in extensive outdoor environments [Ph.D. thesis]*, University of Sydney, Australian Centre for Field Robotics, Sydney, Australia, 2002.

[27] E. Guerra, R. Munguia, Y. Bolea, and A. Grau, "Validation of data association for monocular SLAM," *Mathematical Problems in Engineering*, vol. 2013, Article ID 671376, 11 pages, 2013.

[28] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point RANSAC for EKF-based structure from motion," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '09)*, pp. 3498–3504, St. Louis, Mo, USA, October 2009.

[29] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV '03)*, pp. 1403–1410, Washington, DC, USA, October 2003.

[30] G. Fontana, M. Matteucci, and D. G. Sorrenti, "Rawseeds: building a benchmarking toolkit for autonomous robotics," *SpringerBriefs in Applied Sciences and Technology*, vol. 7, pp. 55–68, 2014.