Information Sciences 279 (2014) 326–337

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A modified stochastic gradient descent algorithm for view-based SLAM using omnidirectional images

David Valiente *, Arturo Gil, Lorenzo Fernández, Óscar Reinoso

Miguel Hernández University, System Engineering Department, 03202 Elche, Spain

ARTICLE INFO

Article history:
Received 25 March 2013
Received in revised form 25 March 2014
Accepted 29 March 2014
Available online 12 April 2014

Keywords:
Mobile robotics
Visual SLAM
Omnidirectional images
SGD

ABSTRACT

This paper describes an approach to the problem of Simultaneous Localization and Mapping (SLAM) based on Stochastic Gradient Descent (SGD) and using omnidirectional images. In the field of mobile robot applications, SGD techniques have never been evaluated with information gathered by visual sensors. This work proposes a SGD algorithm within a SLAM system which makes use of the beneficial characteristics of a single omnidirectional camera. The nature of the sensor has led to a modified version of the standard SGD to adapt it to omnidirectional geometry. Besides, the angular unscaled observation measurement needs to be considered. This upgraded SGD approach minimizes the non-linear effects which impair and compromise the convergence of traditional estimators. Moreover, we suggest a strategy to improve the convergence speed of the SLAM solution, which inputs several constraints in the SGD algorithm simultaneously, in contrast to former SGD approaches, which process only constraint independently. In particular, we focus on an efficient map model, established by a reduced set of image views. We present a series of experiments obtained with both simulated and real data. We validate the new SGD approach, compare the efficiency versus a standard SGD and demonstrate the suitability and the reliability of the approach to support real applications.

1. Introduction

In the field of mobile robot applications, the problem of SLAM is a crucial factor, due to the need for a complete representation of the environment, especially for navigation purposes. The objective of building a map entails considerable complexity, since the map has to be built incrementally, while, the localization of the robot inside it needs to be calculated simultaneously. Generating a reliable and coherent map is even more challenging and laborious when sensor data is affected by noise, and this directly impairs the simultaneous estimation of the map and the path followed by the robot.

To date, SLAM approaches have been differentiated according to several factors, such as the way to estimate the representation of the map, the main algorithm for computing a solution and the kind of sensor to extract information from the environment. For instance, several map representations were obtained thanks to the extensive use of laser data range and sonar [8]. In this area, maps were principally generated following two representation models [16,11], corresponding, respectively, to 2D occupancy grid maps based on raw laser and 2D landmark-based maps focused on the extraction of features, described from laser data measurements.

* Corresponding author. Tel.: +34 96 665 9005; fax: +34 96 665 8979.
E-mail addresses: dvaliente@umh.es (D. Valiente), arturo.gil@umh.es (A. Gil), l.fernandez@umh.es (L. Fernández), o.reinoso@umh.es (Ó. Reinoso).

http://dx.doi.org/10.1016/j.ins.2014.03.122
0020-0255/

More recently, the tendency has turned to the use of visual information by means of digital cameras. Many applications benefit from the use of these sensors, whose characteristics outperform previous sensors such as lasers in terms of the amount of usable information from the environment for building the map. For instance, the approaches that use two calibrated cameras, known as stereo-pairs, in order to extract a set of 3D visual landmarks determined by a visual description [5]. Other approaches simply exploit a single camera to estimate 3D visual landmarks [2,10]. They initialize the coordinates of each 3D landmark relying on the inverse depth parametrization, since there exists a scale uncertainty about the distance to each landmark which cannot be directly retrieved with a single camera. Omnidirectional cameras have also been used alone [15], and some others have even arranged two omnidirectional images, in order to take the best advantage of the wider field of view provided by these cameras.

As important as the kind of sensor and the map representation is the estimation algorithm for a SLAM scheme. It defines the core of the system, as it is responsible for the ultimate solution. Most extensively used are online methods such as EKF [4], Rao-Blackwellized particle filters [11] and offline algorithms, such as, Stochastic Gradient Descent [7].

The combination of data sensors, map representation and the core of the algorithm therefore determines the final effectiveness of a SLAM which seeks reliability and suitability for realistic applications. Great efforts have been made in this field. For example, certain approaches [4,5,3,2,14] have concentrated on estimating of the position of a set of 3D visual landmarks in a main reference system, while, simultaneously, building the map. The main idea lies in the capability of an EKF filter to converge the estimation to an appropriate solution for the SLAM problem. In this same line of EKF usage, [18] has recently proposed a distinctive map representation consisting of a reduced set of image views, determined by their position and orientation in the environment. Such a technique establishes an estimation of a state vector which includes the map and the current localization of the robot at each timestep k. The estimation of the transition between states at k and $k+1$ considers the wheel's odometry as initial estimate, but also the observation measurements gathered by sensors.

Generally, EKF methods are troublesome in the presence of non-linear errors as they have difficulties in maintaining the convergence of the estimation. This situation normally appears in presence of Gaussian errors introduced by the observation measurement, which usually causes data association problems [12]. A visual observation model, such as the omnidirectional, is susceptible to introduce non-linearities and is thus responsible for this kind of errors. By contrast, an offline algorithm such as SGD [1] may deal with this issue caused by non-linearity effects. Similarly, in [20,19,17], parallel approaches are presented to maintain stability in non-linear contexts.

Regarding the basic goals of this study, we present a new visual SLAM approach based on omnidirectional images and sustained by a SGD solver algorithm which helps overcome the harmful effects caused by errors. To achieve this, and depending on the nature of the problem, different aspects have to be taken into consideration so that the research is conducted towards the achievement of new contributions and advantages compared to former applications based on the standard SGD algorithm [13,7,6,1]. Firstly, a map model has to be adapted to the omnidirectional observation. Along the same line, the standard SGD has to be redesigned to be able to work with the omnidirectional geometry of the images, but also considering the nature of the measurement, which lacks scale. This implies that the solution to the problem is not a trivial one. So, the difference between our approach, which uses a different geometrical environment, and all the previous SGD applications, which consider data range observations in a Cartesian measurement system, should be noted. Next, to improve the efficiency of the standard method, in terms of the convergence speed, we propose a modification in the estimation procedure. The traditional models mentioned above, usually process every odometry and observation measurement (denoted as constraints) independently at each iteration step. By contrast, with the aim of finding a valid solution quickly, we propose a strategy based on the simultaneous use of a certain set of information provided by our visual observation measurements. This proposal might appear to be liable to cause an increase in the required computational resources. Nevertheless, we have concentrated on preventing this by updating several stages of the SGD's iterative optimization so as to avoid possible harmful bottleneck handicaps. Therefore, the main expected contributions and advantages of this SLAM approach compared to traditional approaches might be synthesized as it follows:

- An efficient map model established by a reduced set of omnidirectional images.
- A modified SGD solver algorithm adapted to the omnidirectional geometry which is the basis of the proposed SLAM's observation model. Development of the new differential equations related to the observation measurements.
- Improved efficiency of the estimation thanks to the use of simultaneous constraint processing.

The structure of the paper has been divided as it follows: Section 2 depicts the SLAM problem within this framework. Then, Section 3 describes the general specifications of a SGD algorithm, concentrating on the standard SGD. Section 4 details the proposed modification of the standard SGD and the main contributions mentioned above. Next, Section 5 provides both simulated and real data experimental results to validate the model and test its reliability and expected benefits versus traditional methods. Finally, Section 6 analyzes the results to draw a general conclusion.

2. SLAM

A visual SLAM technique is expected to retrieve a feasible estimation of the position of the robot within a certain environment, which also has to be precisely determined by the estimation. In our approach, the map is composed of a set of

omnidirectional images obtained from different poses in the environment, denoted as views. These views do not represent any physical landmarks, as they will consist of an omnidirectional image captured at the pose $x_l = (x_l, y_l, \theta_l)$ and a set points of interest extracted from that image. Such an arrangement, allows us to exploit the capability of an omnidirectional image to gather a large amount of information in a simple image, due to its large field of view. Thus, an important reduction is achieved in terms of the number of variables for estimating the solution.

The pose of the mobile robot at time t will be denoted as $x_v = (x_v, y_v, \theta_v)^T$. Each view $i \in [1, \ldots, N]$ is constituted by its pose $x_{l_i} = (x_l, y_l, \theta_l)_i^T$, its uncertainty P_{l_i} and a set of M interest points p_j expressed in image coordinates. Each point is associated with a visual descriptor $d_j, j = 1, \ldots, M$.

Thus, the augmented state vector is defined as:

$$\bar{x} = \begin{bmatrix} x_v & x_{l_1} & x_{l_2} & \cdots & x_{l_N} \end{bmatrix}^T \tag{1}$$

where $x_v = (x_v, y_v, \theta_v)^T$ is the pose of the moving vehicle and:

$$x_{l_N} = (x_{l_N}, y_{l_N}, \theta_{l_N})$$

is the pose of the N-view that exists in the map.

2.1. Map building

The map building procedure is described by Fig. 1. The exploration task starts navigating the environment at the origin, denoted as A. At this time, the robot captures an omnidirectional image I_A, stored as a view with pose x_{l_A}. While the robot keeps moving towards the first office room, it is able to find correspondences between I_A and the current omnidirectional image, which makes it able to localize itself. Once the robot enters the office room, the appearance of the images varies significantly, so no matches are found between the current image and image I_A. In this case, the robot will initialize a new view named I_B at the current robot's position, which will be used for localization inside the office room. Finally, the robot completes the exploration of the environment as it traverses the different areas of the environment, while acquiring the rest of the necessary views I_C, I_D, I_E, to compose the final map. The number of views initiated in the map depends directly on the kind of environment and its visual appearance. In particular, in Fig. 1 it may be also perceived a synthesis of the localization procedure carried out by the robot, which translates the depicted comparison between I_A and I_E into a single-computation process.

2.2. Observation model

In accordance with the view-based representation recently presented, a new observation model has to be formulated. The versatility of omnidirectional images enables to apply epipolar constraints [9] to extract an observation measurement, which defines the motion transformation between two poses, as seen in Fig. 1. Actually, these poses represent the positions where the robot acquired two specific images. To that effect, only two images, with a set of corresponding points between them, are required to obtain the transformation. The observation measurement may be expressed as:

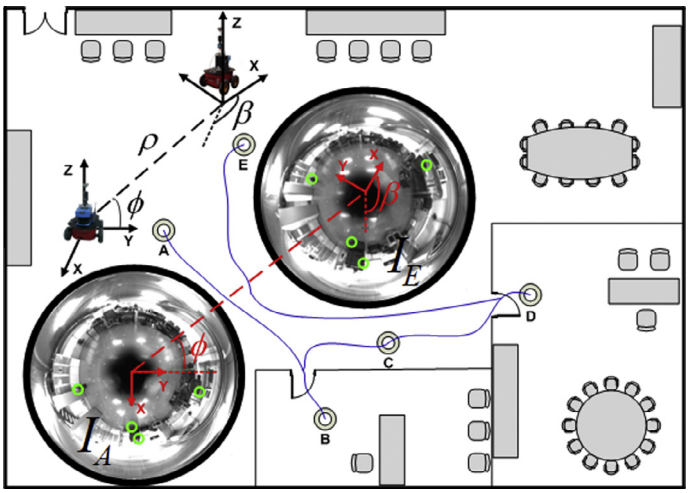

Fig. 1. Map building procedure. The robot starts the exploration at A by acquiring a view I_A. While the robot moves, correspondences are found between I_A and the current image captured at the current robot's pose. When no correspondences are found, the current image is stored as a new view of the map, for instance I_B at B. The procedure ends when the whole environment is represented.

$$z_t = \begin{pmatrix} \phi \\ \beta \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{y_{l_N} - y_v}{x_{l_N} - x_v}\right) - \theta_v \\ \theta_{l_N} - \theta_v \end{pmatrix} \tag{2}$$

where the angle ϕ is the bearing at which the view N is observed and β is the relative orientation between the images. The view N is represented by $x_{l_N} = (x_{l_N}, y_{l_N}, \theta_{l_N})$, whereas the pose of the robot is described as $x_v = (x_v, y_v, \theta_v)$. Both measurements (ϕ, β) are represented in Fig. 1.

3. Standard SGD algorithm

3.1. Specifications

A graph-oriented map is composed by a set of nodes defining the poses traversed by the robot and the landmarks initialized into the map. The state vector s_t encodes this representation through a set of variables which are expressed in the following manner:

$$s_t = [\,(x_0, y_0, \theta_0), (x_1, y_1, \theta_1) \ldots \quad (x_n, y_n, \theta_n)\,] \tag{3}$$

where (x_n, y_n, θ_n) are the 2D coordinates and the bearing in a general reference system. A complementary subset of edges represents the relationships between nodes, by means of either distance measurements generated by the odometry or observations measurements provided by the on-board sensors. Both measurements are commonly known as constraints and denoted as δ_{ji}, where j indicates the observed node, seen from node i. The general objective stated by methods based on standard SGD approaches [13,7] is to minimize the error likelihood, expressed as:

$$P_{ji}(s) \propto \eta \exp\left(-\frac{1}{2}(f_{ji}(s) - \delta_{ji})^T \Omega_{ji} (f_{ji}(s) - \delta_{ji})\right) \tag{4}$$

being $f_{ji}(s)$ a function dependent on the state s_t and both nodes j and i. The difference between $f_{ji}(s)$ and δ_{ji} expresses the error deviation between nodes. Such error term is weighted by the information matrix:

$$\Omega_{ji} = \Sigma_{ji}^{-1} \tag{5}$$

where Σ_{ji}^{-1} is the associated covariance matrix, which considers the uncertainty of the measurements. The assumption of logarithmic notation in (4) leads to:

$$F_{ji}(s) \propto (f_{ji}(s) - \delta_{ji})^T \Omega_{ji} (f_{ji}(s) - \delta_{ji}) = r_{ji}(s)^T \Omega_{ji} r_{ji}(s) \tag{6}$$

being $r_{ji}(s)$ the error determined by $f_{ji}(s) - \delta_{ji}(s)$, which shows its condition of residue. Finally, the global problem seeks the minimization of the objective function which represents the accumulated error:

$$F(s) = \sum_{\langle j,i\rangle \in G} F_{ji}(s) = \sum_{\langle j,i\rangle \in G} r_{ji}(s)^T \Omega_{ji} r_{ji}(s) \tag{7}$$

where $G = \{\langle j_1, i_1\rangle, \langle j_2, i_2\rangle \ldots\}$ defines the subset of particular constraints that define the map, either odometry or observation measurements.

3.2. Estimation

Once the formulation of the problem has been presented, the Stochastich Gradient Descent algorithm must be detailed. The basic goal is to compute in an iterative manner a estimation to achieve a valid solution for the SLAM problem. The basis of a SGD method lies in minimizing Eq. (7) through derivative optimization techniques. The estimated state vector is obtained as:

$$s_{t+1} = s_t + \Delta s \tag{8}$$

where Δs expresses a certain update with respect to s_t, term which is sequentially generated by means of the constraint optimization procedure. It is worth noting that, in a general case, this update is calculated independently at each step by using only a simple constraint, that is to say $\Delta s_n = f(\delta_{ji})$. The general expression for the transition between s_t and s_{t+1} has the following form:

$$s_{t+1} = s_t + \lambda \cdot H^{-1} J_{ji}^T \Omega_{ji} r_{ji} \tag{9}$$

- λ is a learning factor to re-scale the term $H^{-1}J_{ji}^{T}\Omega_{ji}r_{ji}$. Normally, λ takes decreasing values following the criteria $\lambda = 1/n$, where n is the iteration step. This strategy is intended to reach the final solution quickly using large values of λ. When the solution moves close to the optimum, lower values of λ are used, thus preventing the estimation to oscillate around the final solution.
- H is the Hessian matrix, calculated as $J^{T}\Omega J$, and it represents the shape of the error function through a preconditioning matrix to scale the variations of J_{ji}. According to [6], H can be computed:

$$H \approx \sum_{\langle i,j \rangle} J_{ji}\Omega_{ji}J_{ji}^{T} \tag{10}$$

- J_{ji} is the Jacobian of $f_{ji}(s)$ with respect to s_t, $J_{ji} = \frac{\partial f_{ji}}{\partial s}$. It converts the error deviation into a spatial variation.
- Ω_{ji} is the information matrix associated to a constraint. $\Omega_{ji} = \Sigma_{ji}^{-1}$, being Σ_{ji} the covariance matrix corresponding to the observation constraints δ_{ji}.

This scheme updates the estimation by computing the rectification introduced by each constraint at each iteration step respectively. Despite the learning factor to reduce the weight by which each constraint updates the estimation, the procedure may lead to an inefficient method to reach a stable solution, as undesired oscillations may occur due to the stochastic nature of the constraint selection. For this reason, we propose an optimization process which takes into account several constraints in the same iteration. It might be thought that the same drawbacks could arise with the addition of some other inconveniences such as undesired time overloads, as a consequence of the simultaneous processing of several constraints in the same iteration. However, we have modified some calculations at specific stages of the algorithm in order to maintain the time requirements and even reduce them. As a result, we achieved improved convergence ratios in terms of speed. Further details will be provided in the next section.

4. Modified SGD

This section has been intended to explain the main advantages and contributions achieved in this study. The first assumption to consider is the redefined state vector s_t, which will be treated as a set of incremental variables. The pose incremental state is defined as:

$$s_t^{inc} = \begin{bmatrix} (x_0, y_0, \theta_0) \\ (dx_1, dy_1, d\theta_1) \\ \vdots \\ (dx_n, dy_n, d\theta_n) \end{bmatrix} = \begin{bmatrix} (x_0, y_0, \theta_0) \\ (x_1 - x_0, y_1 - y_0, \theta_1 - \theta_0) \\ (x_2 - x_1, y_2 - y_1, \theta_2 - \theta_1) \\ \vdots \\ (x_n - x_{n-1}, y_n - y_{n-1}, \theta_n - \theta_{n-1}) \end{bmatrix} \tag{11}$$

where $(dx_i, dy_i, d\theta_i)$ encode the variation between consecutive poses in coordinates of the global reference system. A global encoding has the main drawback of not being capable to update more than one node and its adjacent ones per constraint. Regarding a relative codification of the state, the problem of non-linearities in J_{ji} arises. By contrast, an incremental state vector allows a single constraint to generate a variation at every pose. In this context, Δs in Eq. (8) affects all poses because the state vector is differentially encoded.

Note that in this approach we are dealing with a visual observation given by an omnidirectional camera. This makes us to adapt the equations defined in the previous section to the case of omnidirectional geometry, as the nature of the constraints are not simply metrical like the odometry constraints. According to (2), given two nodes, the observation measurement allows us to determine a specific motion transformation between them up to a scale factor. Therefore, the omnidirectional measurements and the incremental representation require the reformulation of several terms involved in the estimation. Following this, we detail all the proposed modifications to the terms of the standard SGD, which must be necessarily redefined and recalculated. The complete structure for each derivative is detailed in Appendix A.

- The first modification is referred to $f_{ji}(s)$, differentiating between odometry and visual observation constraints:

$$f_{j,i}^{odo}(s) = \begin{pmatrix} dx_j \\ dy_j \\ d\theta_j \end{pmatrix} + \begin{pmatrix} dx_{j-1} \\ dy_{j-1} \\ d\theta_{j-1} \end{pmatrix} + \cdots + \begin{pmatrix} dx_i \\ dy_i \\ d\theta_i \end{pmatrix} \tag{12}$$

where $(dx_i, dy_i, d\theta_i)$ has been defined in (11). And for the case of the visual observation constraint:

$$f_{j,i}^{visual}(s) = \begin{pmatrix} \phi \\ \beta \end{pmatrix} = \begin{bmatrix} \arctan\left(\frac{dy_j - dy_i}{dx_j - dx_i}\right) - d\theta_i \\ d\theta_j - d\theta_i \end{bmatrix} \tag{13}$$

where β and ϕ are directly computed from the observation measurement [18] model, which expresses the relationship of transformation between two omnidirectional images and the encoded pose of the robot in Eq. (11). Observing Fig. 1 may also help understand the definition of Eq. (13).

- The second modification considers the recalculation of $J_{ji} = \frac{\partial f_{ji}}{\partial s}$, according to the previous reformulation of $f_{ji}(s)$. The importance of considering the indexes of the corresponding nodes, either $j > i$ or $j < i$ must be noted, as the derivatives considerably change its form. Furthermore, as seen above, the dimensions of $f_{ji}(s)$ are different, something which also has to be taken into consideration, as the rest of the terms involved in the SGD algorithm have to be resized.

$$J_{j,i} = \frac{\partial f_{j,i}(s)}{\partial s} = \left[\frac{\partial f_{j,i}(\phi)}{\partial s}, \frac{\partial f_{j,i}(\beta)}{\partial s}\right] \tag{14}$$

- Finally, we suggest the estimation of the new state s_{t+1} by considering several constraints at the same time. We seek greater relevance of the weight of the constraints when searching for the optimal minimum estimation. Obviously, computing more than one constraint at each step leads to a certain overload. By contrast, with this approach, we reduce the expensive estimation of H. In a general case, H is computed whenever a single constraint is introduced, that is to say, as many times as there are constraints. In our case we compute H only once for each subset of constraints introduced simultaneously into the system. Consequently we obtain H in a more efficient manner, thus compensating for possible time overloads. The following example depicts the practical meaning of this concept.

Require: $\delta_{ji} \in C \; \forall \, j, i$, where $C = [c_1, c_2, \ldots, c_b]$ and $c_b = \{\delta_{11}, \delta_{12}, \ldots\}$
  Each c_b represents different subset of constraints δ_{ji} simultaneously processed by the robot.
  t: iteration step
  ϵ: threshold for $F(s)$
  while $F(s) > \epsilon$ **do**
    $t = t + 1$
    for $q = 1{:}b$ **do**
      Extract all δ_{ji} in c_q randomly
      Computing the following terms:
      $f_{ji}(s) = [f_{ji}^{odo}(s), f_{ji}^{visual}(s)]$, J_{ji}, H, Ω_{ji}, and r_{ji}
      $\Delta s_q = \lambda \cdot H^{-1} J_{ji}^T \Omega_{ji} r_{ji}$
      $s_q = s_{q-1} + \Delta s_q$
    end for
    $s_t = s_q + s_{t-1}$
  end while
  return $s_t = [(x_0, y_0, \theta_0), (dx_1, dy_1, d\theta_1), \ldots, (dx_n, dy_n, d\theta_n)]$

---

5. Results

We have carried out three different sets of experiments. Firstly, in Section 5.1 we show SLAM results obtained from simulated data to confirm the validity of the new SLAM approach supported by SGD. We add a comparison of results obtained by our approach with a standard SGD algorithm, like those used in applications like [13,7,6,1]. Finally, in Section 5.2 we present SLAM results using real data acquired by the robot, which have also been compared with a traditional SGD estimator. The equipment consists of a Pioneer P3-AT indoor robot with a firewire 1280 × 960 camera and a hyperbolic mirror. The optical axis of the camera is installed approximately perpendicular to the ground plane, as described in Fig. 1. Consequently, a rotation of the robot corresponds to a rotation of the image with respect to its central point. In addition, we used a SICK LMS range finder in order to compute a ground truth using the method presented in [16].

5.1. SLAM results with simulated data

Confirmation of the convergence of an SLAM algorithm is crucial when a new solver proposal, such as SGD, is introduced. Furthermore, other considerations require evaluation, since the performance of the new method has to deal with a visual observation model, which is a common source of non-linearities.

5.1.1. Experiment 1

Fig. 2(a) presents a random simulation environment of 20 × 20 m, where the robot traverses approximately 300 m. The real path followed by the robot is shown with a continuous line, the odometry is represented with a dash-dotted line, and the estimated solution is shown with a dashed line. A set of views have been placed randomly along the trajectory. The arrangement of these views is controlled by an appearance ratio between images, to assure a realistic placement of each view. A grid

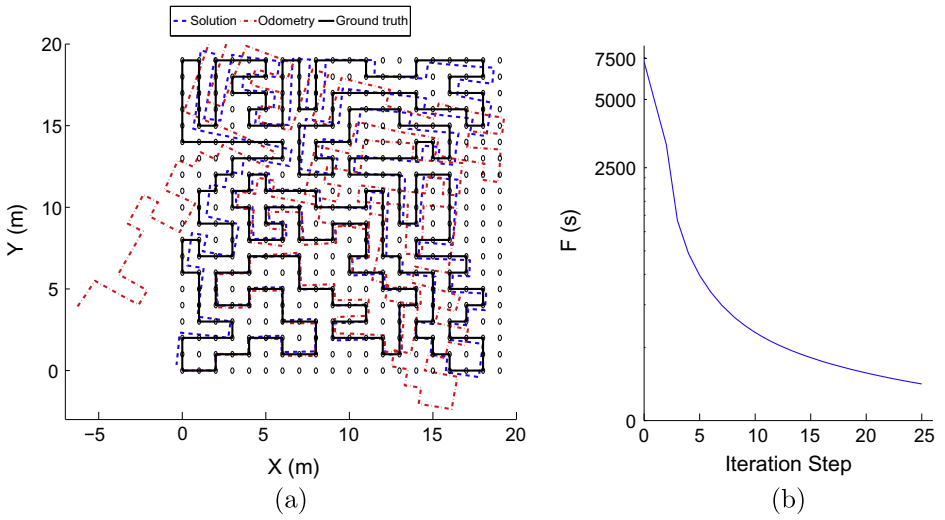

Fig. 2. (a) presents a map obtained by the proposed approach in an environment of 20 times 20 m. The continuous line shows the real path, the dash-dotted line the odometry and the dashed line the estimated solution. (b) Shows the accumulated error probability $F(s)$ along the number of iterations.

of circles represents the possible poses where the robot might move to and gather a new view. The number of iterations of the SGD algorithm is 25. As it can be observed in Fig. 2(a), starting from a noisy odometry estimate, the final estimation has been rectified following the tendency of the real path. Fig. 2(b) shows the decreasing evolution of the accumulated error probability $P_{ji}(s)$ in (4), expressed in logarithmic terms, versus the number of iterations. The reliability of this new approach to work with omnidirectional observations can be confirmed, as it provides a proper solution.

5.1.2. Comparison of results

The following experiments have been conducted in order to compare our approach with the traditional standard SGD in terms of efficiency. We suggest a strategy to introduce several constraints simultaneously into the SGD algorithm. The main goal is to improve the speed by which the method iteratively optimizes until a final solution is achieved. In this sense, we have performed a SLAM experiment, where the robot traverses 50 m through a given environment. Again, the number of views in the map has been randomly placed, by following the same policy explained above. The same experiment has been repeated 200 times using the same series of odometry inputs, in order to provide mean values that express consistent results. The two approaches, ours and the standard SGD algorithm, have been compared. We have modified the number of views N which the robot is able to observe from each pose. The observation range r of the robot has also been varied. Fig. 3 presents results for the accumulated error probability, $F(s)$, being the objective function which the SGD algorithm seeks to minimize. Fig. 3 compares the solution obtained by our approach, drawn with a continuous line, and the solution obtained by the

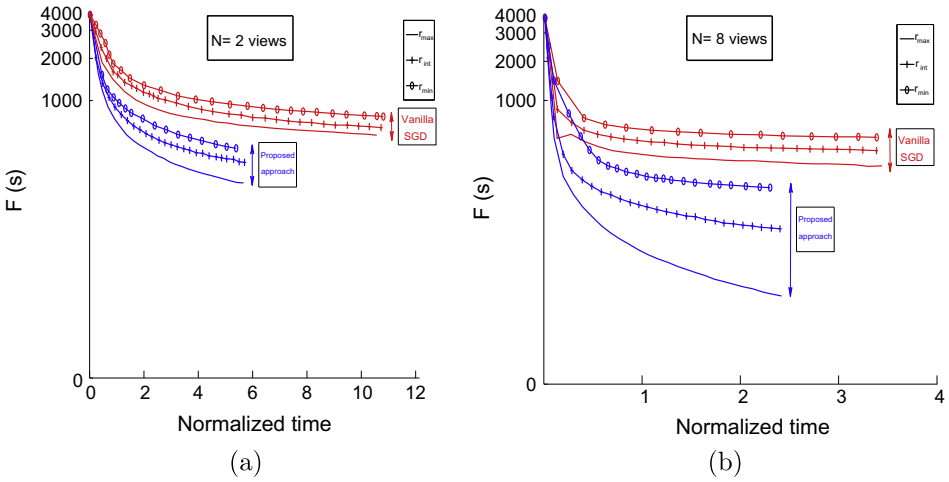

Fig. 3. (a) and (b) Show the accumulated error probability $F(s)$ versus time in a SLAM experiment, when the number of views observed by the robot is $N = 2$ and $N = 8$ respectively. The continuous lines show the results provided by the proposed solution, while the dashed lines show results provided by the standard SGD solution. Different lengths for the observation range are defined: $r_{min}, r_{int}, r_{max}$.

standard SGD algorithm, drawn with a dashed line. Fig. 3(a) and (b) represent $F(s)$ when the robot observes $N = 2$ and $N = 8$ views, respectively. As we are looking for a fair comparison, the x-axis, originally representing iteration steps, has been transformed into a normalized time variable to generate a trustworthy comparison between the two schemes. Please note that the time spent at each iteration step differs from one method to another due to their different convergence speeds. Therefore, in terms of efficiency, it can be shown that the solution provided by our approach outperforms the solution given by a standard SGD in every case, as the decreasing slope of $F(s)$ is clearly steeper. Hence a quicker convergence speed demonstrating a more

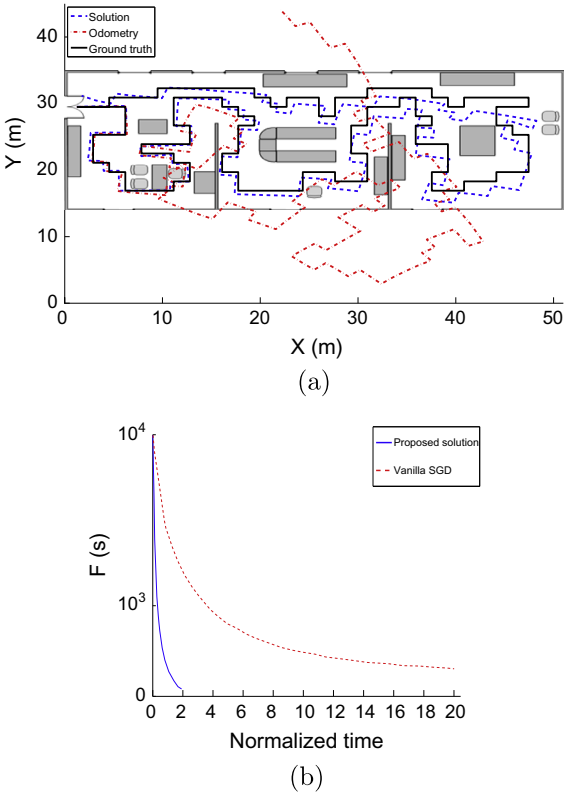

Fig. 4. (a) Shows the SLAM results in an office-like environment of 20 times 50 m. The continuous line shows the real path, the dash-dotted line the odometry and the dashed line the estimated solution. (b) Compares the accumulated error probability $F(s)$ provided by the approach presented in a continuous line and the $F(s)$ provided by the standard SGD in a dashed line.

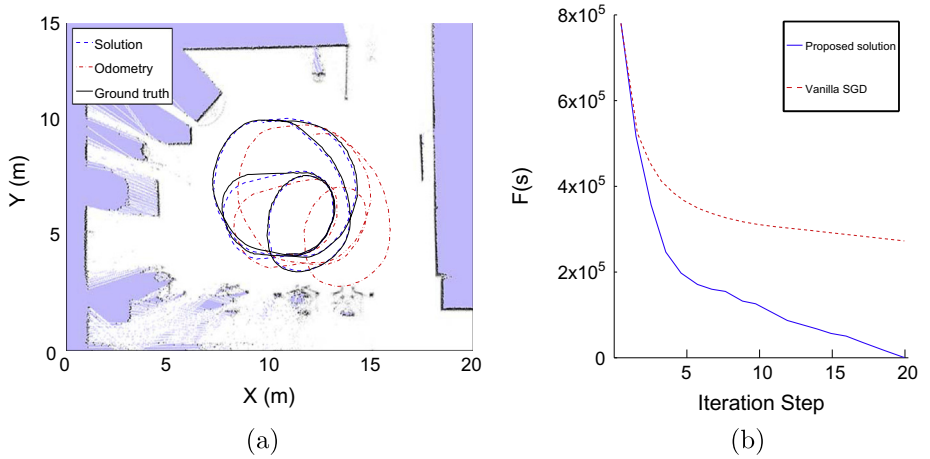

Fig. 5. (a) Shows SLAM results in a real office environment of 15 times 15 m. The continuous line shows the real path, the dash-dotted line the odometry and the dashed line the estimated solution. (b) Shows the accumulated error probability $F(s)$ along the number of iterations for our approach and the standard SGD respectively.

efficient method. This is the main advantage achieved by means of combining several constraints simultaneously at each iteration step, instead of using only one as a traditional SGD used to. The relevance of the observation range of the vehicle r is also notable. As seen in Fig. 3(a) and (b), longer values of r provide a better convergence, compared to shorter r. As the omnidirectional observation is angular, and lacks scale, views seen by the robot at longer distances in the map allow the computation of a more feasible localization. In addition, when the robot is able to observe a higher number of views, the optimum value for $F(s)$ is evidently lower, and is reached quickly, as there are more constraints to compute.

5.1.3. Experiment 2

The purpose of this next experiment is to confirm the favorable results shown in Section 5.1, now dealing with an office-like environment, since it is desirable to emulate a more realistic situation, with obstructions, obstacles, etc. Fig. 4(a) describes the environment of 20×50 m which the robot moves through. The continuous line represents the real path followed by the robot, the dash-dotted line shows the odometry, whereas the estimated solution with our approach is shown by a dashed line. It may be noticed that in only 15 iterations of the algorithm the robot is able to estimate a quiet reliable solution, whose topology follows the real path. On the other hand, the odometry error grows out of bounds. Fig. 4(b) shows a

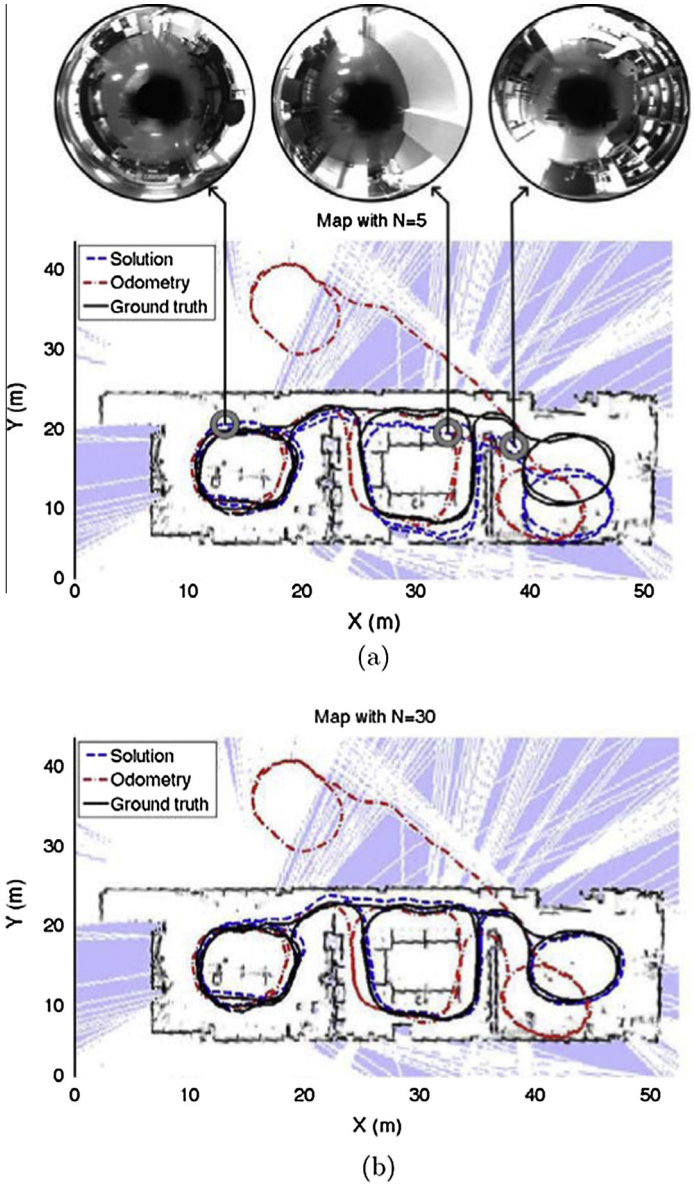

Fig. 6. (a) and (b) show SLAM results in a real office environment of 20 times 50 m, with $N = 5$ and $N = 30$ views observed respectively. The continuous line shows the real path, the dash-dotted line the odometry and the dashed line the estimated solution.

comparison of the evolution of the accumulated error $F(s)$ along the time for both our SGD approach and the standard SGD. Once again, the improved capability in quickly reaching a solution is shown, demonstrating better efficiency. In this particular case, it is worth mentioning that our approach requires a computational cost approximately six times lower than that for a standard SGD to reach an optimum value.

5.2. SLAM results with real data

Having presented the simulation results validating the proposed approach, we carried out a set of experiments with real data. We were seeking for confirmation of the suitability and reliability of the approach in a realistic application such as exploration tasks. Furthermore, we also show comparisons with the standard SGD.

5.2.1. Experiment 3

The first experiment analyzes the behavior of the approach when dealing with one of the most adverse situations, that is to say, when the robot constantly turns around as shown in Fig. 5(a). The real path is shown with a continuous line, the odometry with a dash-dotted line and the estimated solution with a dashed line. This case is seen as one of the worst, as the frequent turns introduce significant noise into the input associated with the odometry. Nevertheless, it should be noted that the estimation converges to a proper solution, which is practically overlapped with the real path, whereas the odometry estimation differs considerably. Fig. 5(b) shows the decreasing tendency of the accumulated error probability $F(s)$ along the number of iterations for both our approach and the standard SGD. Having tested the validity of the main benefits with the previous experiments, the improved efficiency of our approach can now be confirmed in terms of the speed of convergence compared to the standard SGD method. Examinating Fig. 5(b), it can be seen that this approach reaches optimum values for $F(s)$ in less time than the standard SGD. The main advantage in terms of efficiency is therefore shown.

5.2.2. Experiment 4

This last experiment aims to support the beneficial results presented above, which have been compared to traditional SGD approaches. In this case we conducted an experiment in a large environment. Here, the robot moves through a real office of 20×50 m. Again, there are obstacles and obstructions such as doors, walls and office furniture. As seen in Fig. 6 the robot explores the whole environment describing a trajectory of approximately 280m. Moreover, maps with different number of views N have been constructed to study its relevance to the estimation of the solution. Fig. 6(a) and (b) show different results when the robot observes $N = 5$ and $N = 30$ respectively. The real path is drawn with a continuous line, odometry with a dash-dotted line and the estimated solution with a dashed line. Some real views stored in the map have been indicated. Fig. 7 shows the accumulated error probability $F(s)$ for both experiments, expressing it with continuous line for $N = 5$ views and with dashed line for $N = 30$ views. In addition, to demonstrate the improved efficiency of the method, we compare the values of $F(s)$ provided by this approach, in blue, versus that obtained by a standard SGD, in red. In accordance to the

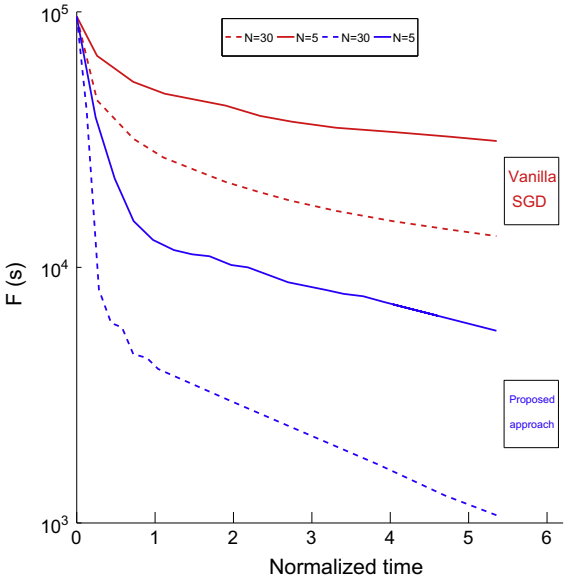

Fig. 7. Accumulated error probability $F(s)$ in a real SLAM experiment versus time. Results obtained for the map showed in Fig. 6(a) with $N = 5$ views, are compared using continuous lines: the continuous blue line represents the proposed approach while the continuous red line represents the standard SGD. Results obtained for the map shown in Fig. 6(b) with $N = 30$ views are compared using dashed lines: the dashed blue line represents the proposed approach whereas the dashed red line represents the standard SGD.

specific topology of the environment, it is confirmed that the larger the number of views N, the more accurate the estimation. Since the robot is able to observe more views, the rectification of the estimation is ensured by a higher number of constraints. Moreover, our approach still reveals the main favorable features compared to the standard SGD, regardless of the value of N. Along the same lines as the previous experiment, the faster convergence speed is proved by observing Fig. 7, where lower optimum values for $F(s)$ are confirmed in considerably less time. This fact shows the greater efficiency of this proposal compared to former SGD techniques.

6. Conclusions

This work has proposed an approach to the visual SLAM problem by introducing a SGD algorithm adapted to omnidirectional observations. The assumption of SGD has been aimed at reducing instabilities and harmful effects which compromise the convergence of the most extended SLAM algorithms such as EKFs. These erroneous effects are mainly consequences of the visual nature of the observation, which is non-linear, and particularly intensified on omnidirectional images. We present a visual SLAM approach which computes a map consisting of a reduced set of omnidirectional views. A single computation of two views allows us to easily retrieve a motion transformation between the poses where the robot captured the views. The standard SGD algorithm has been modified to integrate an unscaled observation model. We propose a more efficient SGD model, which suggests a new strategy designed to exploit the information provided by several constraints simultaneously into the same SGD iteration. We have presented SLAM results with simulated data which validate the combination of SGD with omnidirectional images proposed by this new approach. In addition, we have established a comparison between the results obtained by our approach and those obtained by a standard SGD algorithm. Finally, SLAM results with real data have been presented so as to demonstrate the suitability of the approach and also its efficiency compared to the traditional SGD algorithm.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Science and Innovation under the Project DPI2010-15308 and the FPU scholarship BES-2011-043482.

Appendix A. SGD equations adapted to omnidirectional observations

$$J_{j,i} = \frac{\partial f_{j,i}(s)}{\partial s} = \left[\frac{\partial f_{j,i}(\phi)}{\partial s}, \frac{\partial f_{j,i}(\beta)}{\partial s} \right] \tag{A.1}$$

- $j > i$
  - $k > i$

$$\frac{\partial f_{j,i}(\phi)}{\partial s} = \begin{cases} \frac{\partial f_{j,i}(\phi)}{\partial dx_k} = -\frac{\sum_{k=i+1}^{j} dy_k}{q} = a \\ \frac{\partial f_{j,i}(\phi)}{\partial dy_k} = \frac{\sum_{k=i+1}^{j} dx_k}{q} = b \\ \frac{\partial f_{j,i}(\phi)}{\partial d\theta_k} = 0 \end{cases} \tag{A.2}$$

$$\frac{\partial f_{j,i}(\beta)}{\partial s} = \begin{cases} \frac{\partial f_{j,i}(\beta)}{\partial dx_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial dy_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial d\theta_k} = 1 \end{cases} \tag{A.3}$$

  - $k < i$

$$\frac{\partial f_{j,i}(\phi)}{\partial s} = \begin{cases} \frac{\partial f_{j,i}(\phi)}{\partial dx_k} = 0 \\ \frac{\partial f_{j,i}(\phi)}{\partial dy_k} = 0 \\ \frac{\partial f_{j,i}(\phi)}{\partial d\theta_k} = -1 \end{cases} \tag{A.4}$$

$$\frac{\partial f_{j,i}(\beta)}{\partial s} = \begin{cases} \frac{\partial f_{j,i}(\beta)}{\partial dx_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial dy_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial d\theta_k} = 0 \end{cases} \tag{A.5}$$

$$J_{j,i} = \frac{\partial f_{j,i}(s)}{\partial s} = \begin{Bmatrix} 0 & 0 & -1 \dots & a & b & 0 \dots \\ 0 & 0 & 0 \dots & 0 & 0 & 1 \dots \end{Bmatrix} \tag{A.6}$$

- $j < i$
  - $k > i$

$$\frac{\partial f_{j,i}(\phi)}{\partial s} = \begin{cases} \frac{\partial f_{j,i}(\phi)}{\partial dx_k} = -\frac{\sum_{k=i+1}^{j} dy_k}{q} = a \\ \frac{\partial f_{j,i}(\phi)}{\partial dy_k} = \frac{\sum_{k=i+1}^{j} dx_k}{q} = b \\ \frac{\partial f_{j,i}(\phi)}{\partial d\theta_k} = -1 \end{cases} \tag{A.7}$$

$$\frac{\partial f_{j,i}(\beta)}{\partial s} = \begin{cases} \frac{\partial f_{j,i}(\beta)}{\partial dx_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial dy_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial d\theta_k} = -1 \end{cases} \tag{A.8}$$

  - $k < i$

$$\frac{\partial f_{j,i}(\phi)}{\partial s} = \begin{cases} \frac{\partial f_{j,i}(\phi)}{\partial dx_k} = 0 \\ \frac{\partial f_{j,i}(\phi)}{\partial dy_k} = 0 \\ \frac{\partial f_{j,i}(\phi)}{\partial d\theta_k} = -1 \end{cases} \tag{A.9}$$

$$\frac{\partial f_{j,i}(\beta)}{\partial s} = \begin{cases} \frac{\partial f_{j,i}(\beta)}{\partial dx_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial dy_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial d\theta_k} = 0 \end{cases} \tag{A.10}$$

$$J_{j,i} = \frac{\partial f_{j,i}(s)}{\partial s} = \begin{bmatrix} 0 & 0 & -1 \dots & a & b & -1 \dots \\ 0 & 0 & 0 \dots & 0 & 0 & -1 \dots \end{bmatrix} \tag{A.11}$$

References

[1] C. Berger, Weak constraints network optimiser, in: Proceedings of the International Conference on Robotics and Automation (ICRA), Saint Paul, USA, 2012, pp. 1270–1277.
[2] J. Civera, A.J. Davison, J.M. Martínez Montiel, Inverse depth parametrization for monocular SLAM, IEEE Trans. Robotics 24 (2008) 932–945.
[3] A.J. Davison, Y. Gonzalez Cid, N. Kita, Real-time 3D SLAM with wide-angle vision, in: Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 2004, pp. 117–124.
[4] A.J. Davison, D.W. Murray, Simultaneous localization and map-building using active vision, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 24 (2002) 865–880.
[5] A. Gil, O. Reinoso, M. Ballesta, M. Juliá, L. Payá, Estimation of visual maps with a robot network equipped with vision sensors, Sensors 10 (2010) 5209–5232.
[6] G. Grisetti, C. Stachniss, W. Burgard, Non-linear constraint network optimization for efficient map learning, IEEE Trans. Intell. Transport. Syst. 10 (2009) 428–439.
[7] G. Grisetti, C. Stachniss, S. Grzonka, W. Burgard, A tree parameterization for efficiently computing maximum likelihood maps using gradient descent, in: Proceedings of the Robotics: Science and Systems (RSS), Atlanta, USA, 2007, pp. 1–8.
[8] S. Guadarrama, A. Ruiz-Mayor, Approximate robotic mapping from sonar data by modeling perceptions with antonyms, Inf. Sci. 180 (2010) 4164–4188.
[9] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.
[10] C. Joly, P. Rives, Bearing-only SAM using a minimal inverse depth parametrization, in: Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol. 2, Funchal, Madeira, Portugal, 2010, pp. 281–288.
[11] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, Fast SLAM: a factored solution to the simultaneous localization and mapping problem, in: Proceedings of the 18th National Conference on Artificial Intelligence, Edmonton, Canada, 2002, pp. 593–598.
[12] J. Neira, J.D. Tardós, Data association in stochastic mapping using the joint compatibility test, IEEE Trans. Robotics Automat. 17 (2001) 890–897.
[13] D. Olson, J. Leonard, Fast iterative alignment of pose graphs with poor initial estimates, in: S. Teller (Ed.), Proceedings of the International Conference on Robotics and Automation (ICRA), Orlando, USA, 2006, pp. 2262–2269.
[14] S. Park, S. Kim, M. Park, S.P. Kim, Vision-based global localization for mobile robots with hybrid maps of objects and spatial layouts, Inf. Sci. 179 (2009) 4174–4198.
[15] S.-E. Yu, D. Kim, Image-based homing navigation with landmark arrangement matching, Inf. Sci. 181 (2011) 3427–3442.
[16] C. Stachniss, G. Grisetti, D. Haehnel, W. Burgard, Improved Rao-blackwellized mapping by adaptive sampling and active loop-closure, in: Proceedings of the Workshop on Self-Organization of Adaptive behavior (SOAVE), Ilmenau, Germany, 2004, pp. 1–15.
[17] Y.T. Sun, C.-H. Wang, C.C. Chang, Switching T–S fuzzy model-based guaranteed cost control for two-wheeled mobile robots, Int. J. Innov. Comput. Inf. Control 8 (2012) 3015–3028.
[18] D. Valiente, A. Gil, L. Fernández, O. Reinoso, View-based maps using omnidirectional images, in: Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol. 2, Rome, Italy, 2012, pp. 48–57.
[19] L. Wu, D.W.C. Ho, Fuzzy filter design for It stochastic systems with application to sensor fault detection, IEEE Trans. Fuzzy Syst. 17 (2009) 233–242.
[20] R. Yang, H. Gao, P. Shi, Delay-dependent robust H control for uncertain stochastic time-delay systems, Int. J. Robust Nonlinear Control 20 (2010) 1852–1865.