Robotics and Autonomous Systems 58 (2010) 991–1002

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Real-time hierarchical stereo Visual SLAM in large-scale environments☆

David Schleicher*, Luis M. Bergasa, Manuel Ocaña, Rafael Barea, Elena López

Department of Electronics, University of Alcala, Alcalá de Henares, 28805 Madrid, Spain

ARTICLE INFO

Article history:
Received 3 December 2008
Received in revised form
26 March 2010
Accepted 31 March 2010
Available online 24 April 2010

Keywords:
Mobile robots
Stereo vision
Tracking

ABSTRACT

In this paper we present a new real-time hierarchical (topological/metric) Visual SLAM system focusing on the localization of a vehicle in large-scale outdoor urban environments. It is exclusively based on the visual information provided by a cheap wide-angle stereo camera. Our approach divides the whole map into local sub-maps identified by the so-called fingerprints (vehicle poses). At the sub-map level (low level SLAM), 3D sequential mapping of natural landmarks and the robot location/orientation are obtained using a top-down Bayesian method to model the dynamic behavior. A higher topological level (high level SLAM) based on fingerprints has been added to reduce the global accumulated drift, keeping real-time constraints. Using this hierarchical strategy, we keep the local consistency of the metric sub-maps, by mean of the EKF, and global consistency by using the topological map and the MultiLevel Relaxation (MLR) algorithm. Some experimental results for different large-scale outdoor environments are presented, showing an almost constant processing time.

1. Introduction

Real-time Simultaneous Localization and Mapping (SLAM) is a key component in robotics and it has seen significant progress in the last decade [1–3]. The interest in camera-based SLAM has grown tremendously in recent years. Cameras have become much more inexpensive than lasers, and also provide texture rich information about scene elements at practically any distance from the camera. Currently, the main goal in SLAM research is to apply consistent, robust and efficient methods for large-scale environments in real-time.

Traditionally, vision researchers have concentrated on reconstruction problems focusing on the so-called *Structure From Motion* (SFM) techniques. This methods estimate the ego-motion from frame to frame feature matching and perform global estimation optimization by means of the method known as *bundle adjustment* [4]. Because of its implementation is carried out essentially offline [5], these methods are not well suited for consistent localization over arbitrarily long sequences in real time. Some methods make use of bundle adjustment techniques but only to a reduced set of keyframes of the sequence. Thus, vehicle poses associated to these keyframes are calculated and locally optimized. In [6] a real time local bundle adjustment method is presented, which shows accurate vehicle poses and medium size environment reconstruction in real time. This method, however, only estimate a very sparse set of poses to be able to process a large amount of landmarks. Also, the fact that a monocular sensor is used, implies a prior knowledge of the initial environment.

One of the most popular methods to solve the SLAM problem is the Extended Kalman Filter (EKF). As it is well known, the EKF implementation is limited by the complexity of the covariance matrix calculation, which increases quadratically in large-scale maps as a function of the landmarks introduced into the filter $O(n^2)$. To deal with that problem, in the last years the so-called FastSLAM algorithm was presented [7]. It recursively estimates the full posterior distribution over the robot's pose and landmark locations by using a particle filter to model multiple path hypotheses. It has been widely applied [8,9], however as long as the environment becomes larger, the processing time needed to calculate the different hypotheses increases dramatically.

If we focus on the EKF based algorithms, two main different approaches are taken to face the complexity problem. On one hand, several methods try to modify the intrinsic principles of the EKF regarding the way of the covariance matrix is computed [10,11]. Most of them achieve to reduce the problem to a linear complexity order $O(n)$. Instead of intrinsically modifying the filter, some other methods have focused on facing the problem of global localization and mapping by dividing the map into smaller ones using a metric [12–14] or topological approach [15,16]. They both use detailed local maps, but the sub-maps employed in the metric approach do not maintain a topological structure of an environment as in the hybrid or topological/metric approach.

☆ This work was supported in part by the Spanish Ministry of Education and Science (MEC) under grant TRA2005-08529-C02 (MOVICON Project) and grant PSE-370100-2007-2 (CABINTEC Project) as well as by the Community of Madrid under grant CM: S-0505/DPI/000176 (RoboCity2030 Project).

* Corresponding author. Tel.: +34 666244074; fax: +34 918856591.
E-mail addresses: dsg68818@telefonica.net, dsg68818@gmail.com (D. Schleicher), bergasa@depeca.uah.es (L.M. Bergasa), mocana@depeca.uah.es (M. Ocaña), barea@depeca.uah.es (R. Barea), elena@depeca.uah.es (E. López).

0921-8890/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2010.03.016

Our work relies on the topological/metric philosophy using local maps in order to represent the world and locate within. Our approach basically generates a series of local sub-maps taken on an equally-spaced basis (low level SLAM). Each of them is composed of a number of visual landmarks precisely taken, and is handled by using a standard EKF. A topological map along with local metric sub-maps is built (high level SLAM). The topological map is a graph-like map consisting of vertices and edges. Each vertex represents a topological place, a robot pose that we call *fingerprint*, and includes a local metric sub-map. If a robot is traveling between two vertices, an edge is inserted to connect these two vertices, which represents a link between these two poses. Meanwhile, the edges store transformation matrices and uncertainties to describe the relationship between connected vertices. Using this hierarchical strategy of two levels, on one hand we keep the local consistency of the sub-maps by mean of the EKF and, on the other hand, we keep global consistency by using the topological level and the MultiLevel Relaxation (MLR) method of Frese et al. [17]. The MLR algorithm determines the maximum likelihood estimate of all vehicle vertices along the whole path. Vertex corrections are transmitted to the landmarks of their corresponding sub-maps.

Our final goal is the autonomous outdoor navigation of a vehicle in large-scale environments where a GPS signal does not exist or is not reliable (tunnels, urban areas with tall buildings, mountainous forested environments, etc.). Our research objective is to develop a robust localization system, based on SLAM using only a cheap stereo camera, able to complement a standard GPS sensor, for vehicle navigation. Then, our work is focused on real-time localization as the main output of interest. A map is certainly built, but it is a sparse map of landmarks optimized toward enabling localization. Our hierarchical proposal of two levels (topologic and metric) works well in large-scale environments, producing topological correct and geometrical accurate sub-maps at minimal computational cost. On the other hand, the topological level facilitates the path planning strategies, fusion with the GPS information and the future generalization of the system to a multi-vehicle SLAM.

2. Related work

In [3], Davison presented an impressive work of real-time 3D visual SLAM carried out by using a hand-held single camera. It was the main basis of our research. In his recent paper [18] Davison presented a revision of his method called MonoSLAM. MonoSLAM is an EKF SLAM system, and cannot be used to map large environments. To solve the covariance complexity problem, several strategies have been developed in recent years. We will focus our study on the submapping strategies.

One possible solution to the large scale problem is the *Metric–Metric* approach, which faces it dividing the whole map into smaller ones using a high metric level approach over the metric sub-maps. One of the first methods that applied techniques for map splitting was presented by Tardós et al. [13]. They intend to map and locate a robot within by using sonar measurements. One of their main contributions was to create local sub-maps applying EKF within them. The independent sub-maps are joined afterwards by using compositions. An important problem of this method and the Hierarchical Visual SLAM one is that local maps must be statistically independent. This impedes sharing valuable information between local maps. A solution for this problem has been recently published by the authors in [19] where a Conditionally Independent Divide and Conquer SLAM is proposed. In order to extend the MonoSLAM method to larger environments a Hierarchical Visual SLAM is presented in [12]. A single camera is used in both these systems, and thus scale unobservability is a fundamental limitation in both. In either case, the scale must be fixed by observing known objects to avoid drift in scale over time. Hierarchical Visual SLAM can be used for large scale mapping because it divides the global map in local sub-maps of limited size, achieving almost constant time execution. One of the last contributions is the work presented in [14]. A 6DOF stereo-in-hand system, based on the commercial Bumblebee stereo system, is used to capture visual landmarks, but this time they are classified as either nearby or far. Depending on this, information provided by the stereo pair will be either complete location or just angular information of the landmark relative to the camera. This methodology is an evolution of their previous monocular version [19]. An EKF sub-map strategy is also applied here. Results show an accurate mapping and loop closing over relatively large environments. However, due to the lateral movement of the camera, a continuous matching philosophy is imposed in order to reduce time frames to detect loop closing situations. The use of a relatively close range camera system does not make it suitable for very large and open-spaced environments, where most of the landmarks will be too far. Also, real-time behavior is not completely achieved.

Another alternative to solve the large scale problem is to use a high topological level approach over the metric sub-maps, which leads to the *Topological-Metric* methods. In [20] they present the *Decoupled Stochastic Mapping* (DSM), where a global map is divided into smaller *cells* containing parts of the global one. All landmarks and vehicle poses are referred to the global frame in any of the cells. Crossing from one cell to another implies an information transfer solved using uncertainty inflation methods, which are questionable. Also, the closing loop optimization issue is not addressed on this method. *Hierarchical Local Maps* (HLM) method is presented in [21]. It consists in a hierarchical set of sub-maps locally referenced in this case. Adding a new sub-map implies storing the local vehicle pose and covariance at that moment. All the estimates are stored in a coupling tree, where relations between any of the sub-maps can be calculated using *coupling summation* formulas. One of the main disadvantages is the fact that coupling estimates of all sub-maps remain static throughout. This implies that no uncertainty reduction can be performed when closing some loops. The *Constrained Relative Submap Filter* (CRSF) presented in [22] is essentially equal to HLM, but introduces improvements on the way coupling estimates are stored, which allow, in case that the vehicle returns back to the previous sub-map, reinitializing the vehicle estimation using geometric constraints. This permits reducing the uncertainty of the subsequent sub-map as the previous one also converges. However, due to the monotonic linkage between sub-maps, no global optimization is performed in case of loop closing situations. *Network Coupled Feature Maps* (NCFM) presented in [23] is based on CRSF as well. However NCFM does not restrict coupling estimates to monotonic linkages, allowing further optimization in loop closing situations. One advantage of the method is that it allows optimizing different sub-maps couplings when vehicle covers boundary regions between these sub-maps. This approach implies to have a relatively dense grid of sub-maps strongly overlapped to exploit this advantage, and to be able to reduce global uncertainty. It also requires a robust data association method to relate visual landmarks between adjacent sub-maps in the case of visual SLAM systems. The approach of Eade and Drummond [24] is based on the NCFM method. It consists in a set of interconnected nodes containing Kalman filter map estimates. Map states and uncertainties are computed in their local frames. To reduce linearization errors, measurements are expressed using the inverse depth representation. Edges store the constraints between nodes defined by a similarity transform, which due to the monocular implementation, includes the scale information. The active node is selected based on the visible available landmarks

and the estimation of the measurement model's linearity. The inverse depth implementation shows to improve measurement linearization with limited displacements compared to landmarks distances. However this is true, due to the camera's movement configuration, which implies mainly lateral parallel displacements, keeping landmarks depths almost constant. This assumption is also made in [14]. On the other hand, the algorithm is well suited for indoor environments with strong relations between different regions (i.e. nodes), but is not expected to improve significantly the estimation in large outdoor environments. In [25] a two level hierarchical approach to the SLAM problem is presented. It defines a local level where the robot is located relative to a local reference frame. Then, a global level maintains a topological structure of the environment, where nodes represent the local reference frames of the local level. To implement it, laser scans are used to detect walls, corners, etc., which will be identified as 2D features. Because of that implementation, to detect loop closings a relocation algorithm, based on the structure of the mapped environment is used. This method, however is prone to fail in the case of highly symmetric environments, which is typical in urban scenarios. It has been tested in medium size environments up to 350 m long and at low speed (1.62 km/h average speed). The map optimization time can reach up to 680 ms with a reduced number of features. As no parallelization between global and local levels is carried out, it is expected that real time implementation is not feasible. On the other hand, the way that shared features between local maps are managed makes the system more suitable to highly interconnected environments, like indoors corridors, rooms, etc.

In [26] they present an almost real-time system based on a stereo camera pointed to the floor and an inertial unit. The main problem comes from the reduced field of view of the system. It implies a low reliability of loop closing situations where a highly repetitive texture is captured.

A third alternative to face the large scale SLAM problem is to use only *topological* maps without sub-maps associated to their vertex. These maps lack the details of the environments but they can achieve good results for certain applications. In [27] a minimalist visual SLAM for large-scale environments is presented. The approach is based on a graphical representation of robot poses and links between the poses based on odometry and omni-directional image similarity. A MLR algorithm is used to generate a globally consistent map. For the future they plan to include a thorough run-time evaluation, to substitute the omni-directional camera with a standard one and to incorporate vision-based odometry. Another approach is presented in [28], where a topological map capturing and storing images frame by frame and comparing them with previous ones is built. The system relies on SIFT descriptors efficient matching and storing scheme. In spite of its efficiency, it ends on a computational cost exceedance when too many images have been captured. In [29] they use SIFT descriptors as well to build an appearance based topological map. As it is only topological, no ego-motion information is obtained from the metric point of view. The main contribution is the way that large amount of keyframes are managed to distinguish whether the vehicle visits new places or revisits old ones (i.e. closing a loop). The probabilistic point of view for managing keyframes matching is based on the probability that two different image views come from the same place. This estimation depends also on individual properties of the keyframe, such as the pattern repetitiveness of the image, i.e. how well correlated the SIFT descriptors are within the image. This is quite interesting in cases where the field of view is usually narrow (the camera is pointing laterally to the vehicle's displacement), and so image texture richness tends to be low (walls, trees, etc.). We point our camera to the front of the movement, using also wide angle lenses, so we do not usually face that problem.

To choose one of the three main alternative approaches, we take into account that, on one hand, although Metric–Metric methods provide accurate estimations they do not keep a topological structure that helps on a global optimization in large scale environments as well as path planning techniques for navigation purposes. Topological approaches do not provide accurate information of vehicle state estimations instead. Therefore, our proposal to solve the large-scale problem is based on the hierarchical topological-metric approach and it resembles the NCFM algorithm, but instead of obtained inter-node links among sub-maps using shared map features, we calculate them using the vehicle's trajectory and loop closures. The main contributions of our method compared to more relevant topological-metric proposals presented in this section can be summarized in a more robust data association strategy for large loop closing based on SIFT fingerprints and a simpler node relations management, well suited for large outdoor urban environments. Also, thanks to the use of stereovision, a correct scale estimation of the map is maintained even before closing loops or revisiting places. Our method allows, as well, a continuous global uncertainty estimation of the vehicle at any time. All of this is shown to work in real time on large covered paths with a negligible increase on the processing time as new landmarks are added to the map.

This paper is organized as follows: the general structure of the system is described in Section 3. Section 4 presents the low level SLAM implementation and Section 5 the high level SLAM. In Section 6 a large set of results is given to show the behavior of our system. Section 7 contains our conclusions and future work. This work relies on previous papers presented by the authors on two conferences up to this time [30,31].

3. Implementation

This paper presents a real-time SLAM method for large-scale outdoor environments based only on stereo-vision. To deal with the covariance matrix growing problem, we divide the global map into local sub-maps. Each of these sub-maps has its own metric SLAM process, independent of the other sub-maps. Over these local sub-maps we define a higher topologic SLAM level that relates them keeping the global map consistency.

The system is based on a stereo wide-angle camera mounted on a mobile vehicle. For each local sub-map, several visual landmarks are sequentially captured, using the Shi and Tomasi operator (see [30]), and introduced on an EKF filter in order to model the probabilistic behavior of the system. A measurement model is used for landmark perception and a motion model is implemented for the dynamic behavior of the vehicle, as shown in Fig. 1 *left*. The use of a stereo camera to identify and track features associated to the landmarks allows their direct position calculation. It also avoids both the needing of *a priori* information of the environment as well as scale assumptions. All these tasks are carried out from the metric point of view within the so-called "low level SLAM".

We present a hierarchical SLAM implementation, which adds an additional processing level called "high level SLAM" to the explained "low level SLAM". The whole map is divided into independent local sub-maps identified by *fingerprints*, represented as arrows within circles (see Fig. 1 *right*). These fingerprints store the vehicle's pose at the moment of the sub-map creation and define its local reference frame. The sub-map generation is performed periodically in space so, after a certain covered section of the path, a new sub-map is created and a fingerprint is associated to it. If the vehicle is traveling between two fingerprints, an edge is inserted to connect these two vertices, which represents a link between two poses. Meanwhile, the edges store transformation matrices and uncertainties to describe the relationship between connected fingerprints. The decomposition of the global map into local sub-maps simplifies the problem of map optimization in large-scale environments. This optimization

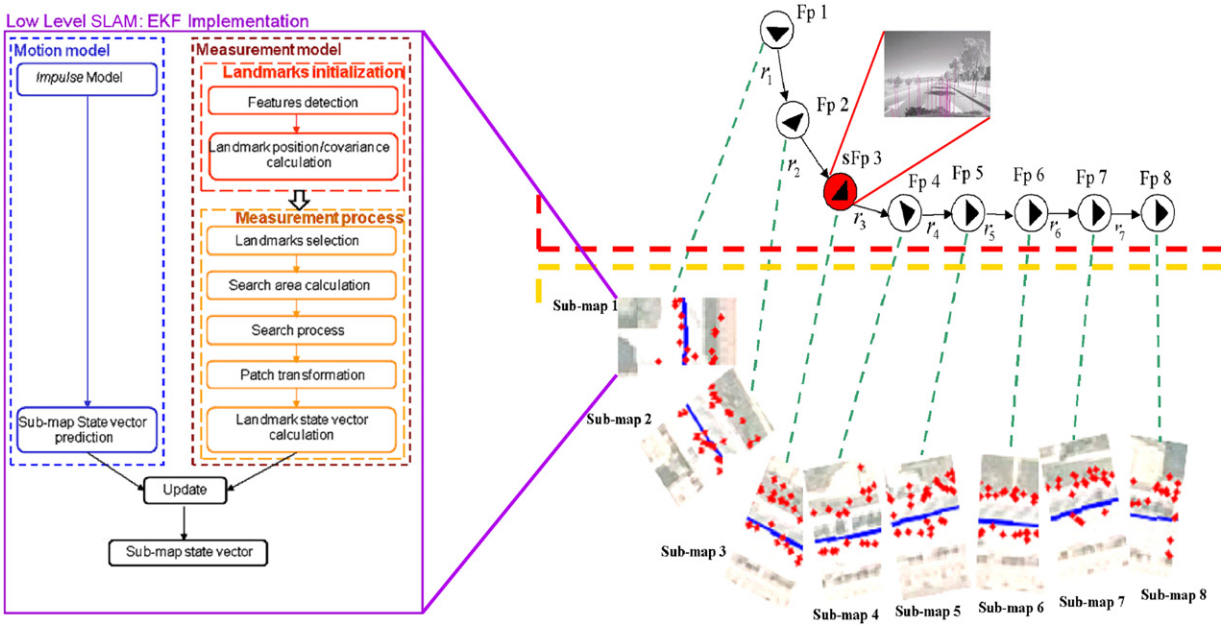

Fig. 1. *Left.* Low level SLAM tasks carried out within each sub-map. *Right.* General architecture of our two hierarchical SLAM levels. Each sub-map has an associated fingerprint.

is carried out at the global map level using an efficient method called MLR [17]. Modifications on the fingerprints as consequence of the optimization are directly transferred to the local sub-maps.

To optimize the loop-closing detection, when a significant vehicle turn is detected, an additional fingerprint called *SIFT fingerprint* is taken. This adds to the vehicle's pose some visual information to identify the place where it was taken. Matching between the previously captured *SIFT fingerprints*, within an uncertainty area, and the current one is carried out to detect pre-visited zones. In case of positive matching, a loop-closing is detected and the topological map is corrected by using the MLR algorithm [17] over the whole set of fingerprints. The MLR determines the maximum likelihood estimate of all fingerprint poses. After that, landmarks of each sub-map are corrected as a function of the correction applied to its associated fingerprint.

4. Method: low level SLAM

This level implements all the algorithms and tasks needed to locate and map the vehicle on its local sub-map. For clarity reasons the sub-map notation is omitted, so it is assumed a unique sub-map for the low level SLAM implementation.

4.1. Extended Kalman filter application

The low level state vector is defined as $X = (X_v \quad Y_1 \quad Y_2 \quad \cdots)^T$, which is composed by the vehicle state vector $X_v = (X_{\text{rob}} \quad q_{\text{rob}} \quad v_{\text{rob}} \quad \omega_{\text{rob}})^T$ plus all local landmarks on the sub-map Y_i. Because of the employed *motion model*, which will be explained later, linear and angular speeds are added to the vehicle state vector. A vehicle coordinate system has been set as the camera frame. On this equation, X_{rob} is the 3D position of the camera relative to the local frame, $q_{\text{rob}} = \begin{pmatrix} q_0 & q_x & q_y & q_z \end{pmatrix}^T$ is the orientation quaternion, v_{rob} is the linear speed and ω_{rob} is the angular speed.

The EKF is applied in the standard form, as explained in [30]. The overall filter process is shown in Fig. 1 (*left*).

4.2. Motion model

To build a motion model for a camera mounted on a mobile vehicle using only visual information, a practical solution is to apply the so-called *impulse model*. This assumes constant speed (both linear and angular) during each time step and random speed changes between steps in the three directions. Some restrictions have been applied to adapt the 6DOF generic model to the vehicle's movement dynamics. According to this model, to predict the next state of the camera the function shown in (1) is applied. The term $q[\omega_{\text{rob}} \cdot \Delta t]$ represents the transformation of a 3 component vector into a *quaternion*. Assuming that the map does not change during the whole process, the absolute feature positions Y_i should be the same from one step to the next one. This model is subtly effective and gives the whole system important robustness even when visual measurements are sparse.

$$f_v = \begin{pmatrix} X_{\text{rob}} + v_{\text{rob}} \cdot \Delta t & q_{\text{rob}} \times q\,[\omega_{\text{rob}} \cdot \Delta t] & v_{\text{rob}} & \omega_{\text{rob}} \end{pmatrix}^T. \quad (1)$$

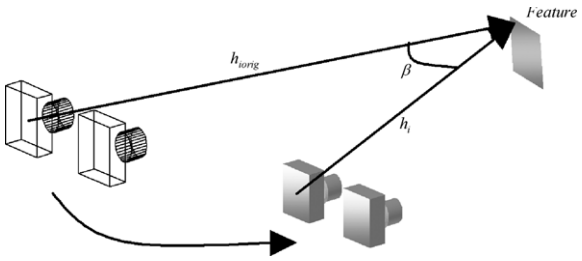

Fig. 2. Original and current feature measurement vectors.

4.3. Measurement model

Visual measurements are obtained from the "visible" feature positions. We define each individual with a 3 component *measurement prediction* vector h_i as the corresponding 3D feature position relative to the left camera frame, which is selected as the reference. To choose the features to measure, some selection criteria have to be defined. These criteria will be based on the feature *visibility*, that is, whether its appearance is close enough to the original one (when the feature was initialized). This is based on the relative distance and point of view angle respect to the one at the feature initialization phase (see Fig. 2).

The first step is to predict the measurement vector h_i (*measurement vector prediction*). To look for the actual measurement vector z_i (*actual measurement vector*), we have to define a search area on the projection images. This area will be around the projection points of the predicted measurement h_i on both *left* and *right* images: $U_L : (u_L, v_L)$, $U_R : (u_R, v_R)$. To obtain the image projection coordinates, first we apply the simple "pin-hole" model and then it is distorted using the radial and tangential distortion models, which are detailed in [30]. To obtain z_i we need to solve the inverse geometry problem, applying the distortion models as well.

Regarding the search areas, they will be calculated based on the uncertainty of the feature's 3D relative position, which is called the *innovation covariance* S_i. As we have two different image projections, S_i needs to be transformed into the projection covariance P_{U_L} and P_{U_R} using Eq. (2).

$$P_{U_L} = \frac{\partial U_L}{\partial h_i} \cdot S_i \cdot \left(\frac{\partial U_L}{\partial h_i}\right)^T; \qquad P_{U_R} = \frac{\partial U_R}{\partial h_i} \cdot S_i \cdot \left(\frac{\partial U_R}{\partial h_i}\right)^T. \tag{2}$$

These two covariances define both elliptical search regions, which are obtained taking a certain number of standard deviations (usually 3) from the 3D Gaussians. Once the areas, where the current projected feature should lie, are defined, we can look for them. At the initialization phase, the left and right images representing the feature *patches* are stored. Then, to look for a feature patch, we perform normalized *sum-of-squared-difference correlations* across the whole search region. We scale and rotate the landmark patch according to the current estimate of camera pose, relative to the pose in which the patch was acquired. Therefore, the patch appearance is warped using the *Patch Adaptation* method described in [30]. This helps on the search correlations phase in the sense of extending the tracking of the patch.

In our application, the camera provides a baseline of $T_{\text{int}} = 400$ mm. We do not make any explicit differentiation between near and far landmarks, as it is done in [14]. However, our method implicitly does that. Far landmarks provide more useful information when the vehicle turns and near landmarks when the vehicle goes straight ahead. The reason is due to the innovation covariance S_i, which at the end provides the weight of each landmark within the filter. In straight movements distant landmarks appear to be almost static, i.e., their innovation from frame to frame is relatively low. However, on the vehicle turns the innovation on distant landmarks is higher, increasing their weights in that situation.

4.4. Feature initialization

The selected criteria to initialize new landmarks are to maintain always at least 5 visible features and 4 successfully measured features, allowing the initialization of 1 feature per frame. Then, when a new feature initialization needs to take place, making use of the Shi and Tomasi operator, its corresponding patch will be searched within a rectangular area randomly located on the left camera image. To obtain the right image feature correspondence we search over the *epipolar line*, restricted to a certain segment around the estimated right projection coordinates. The detailed implementation is described in [30].

In [12] the authors make use of the *joint compatibility branch and bound* (JCBB) [19] outlier rejection technique when measuring landmarks in a single camera 3D SLAM. In our case we use a stereo camera, then the uncertainty in landmark position estimation is much reduced since its creation, reducing the search over large uncertainty areas at any time. At the time of capturing new landmarks we use the epipolar restriction as well as an additional restriction over the epipolar line, avoiding capturing too close landmarks. This clearly reduces the possibility of mismatches. On the other hand, our system is clearly designed to be mounted onboard vehicles within urban areas. The viewpoint direction is always pointed to the front of vehicle's movement having a wide field of view. Therefore it is very unlikely to see highly repetitive textures.

5. Method: high level SLAM

Our SLAM implementation adds an additional topological level, called high level SLAM, to the explained low level SLAM in order to keep global map consistency with almost constant processing time. This goal is achieved by using the MLR algorithm over the so-called *Fingerprints*. Therefore, the global map is divided into local sub-maps identified by the mentioned fingerprints. There are two different classes of fingerprints: *Ordinary Fingerprints* and *SIFT fingerprints*.

The first ones are denoted as $FP = \{fp_l | l \in 0 \cdots L\}$. Their purpose is to store the vehicle local pose $X_{\text{rob}}^{fp_l}$ and local covariance $P_{\text{rob}}^{fp_l}$ relative to the previous fingerprint, i.e., the reference frame of the current sub-map. To define the sub-map size we take into account two main aspects: one is related to the non linearity problem. It is well known that EKF linearization can be assumed only within limited size environments. To cope with that problem we limit the size of the sub-maps to keep the linearization error low enough, as explained in [24]. Also, we found that keeping a constant size in terms of the path covered we obtain better consistency of the results on the high level global map reconstruction. The other aspect is to keep the system under a real time constraint. This implies a limit in the number of landmarks processed on the low level filter. We experimentally found that processing a map with up to 60 landmarks per sub-map we are below the limit. Therefore, we found a suitable sub-map size as 10 m of path covered, so each 10 m a new ordinary fingerprint will be taken.

The second class of fingerprints is a sub-set of the first ones, denoted as $SF = \{sf_q \in FP | q \in 0 \cdots Q, Q < L\}$. The additional functionality is to store the visual appearance of the environment at the moment of being obtained. That is covered by the definition of a set of *SIFT features* associated to the fingerprint, which identifies the place at that time $YF^q = \{Yf_m^q | m \in 0 \cdots M\}$. These fingerprints are taken only under the condition of having a significant change on the vehicle trajectory (see Fig. 3). This change is defined in 2 steps: first the vehicle must have an orientation change $\Delta\theta_1 \geq \gamma_{\max}$ within a time gap. Second, to obtain the most stable point of view every time we revisit the same place, we wait for the SIFT fingerprint capture until the orientation variation falls below a threshold level $\Delta\theta_2 \leq \gamma_{\min}$. The orientation angle can be easily obtained from the quaternion. Both the limits and the time gap have been set after testing several urban environments, avoiding the capture of SIFT fingerprints just for random slight vehicle movements within the road, while capturing them at singular points within the map. Using this approach, theoretically we could face the situation of covering a very large loop, where no obvious turns are made and no SIFT fingerprints would be detected. Our approach is based on the assumption of semi-structured environments with singular identifiable places from the trajectory changes point of view, and such a case would not take place in common urban environments.

Each time a new SIFT fingerprint is taken, it is matched with the previously acquired SIFT fingerprints within an uncertainty search region. This region is obtained from the vehicle global covariance P_{rob}^G because it keeps the global uncertainty information of the vehicle. If the matching is positive, it means that the vehicle is in a previously visited place and a *loop closing* is identified. Then, the MLR algorithm is applied in order to determine the maximum likelihood estimate of all fingerprint poses. Finally, fingerprint corrections are transmitted to their associated sub-maps.

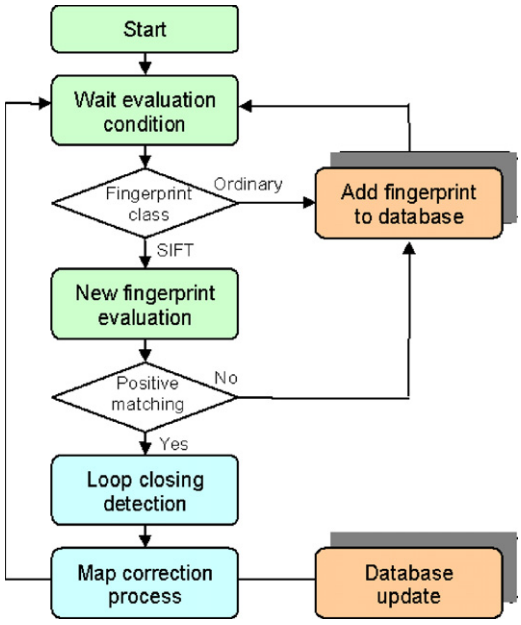

Fig. 3. High level map management.

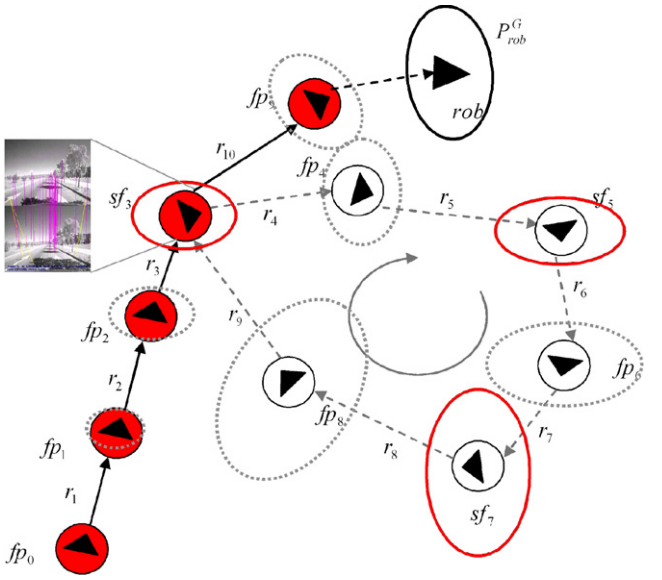

Fig. 4. Representation of the vehicle's global uncertainties P^G_{rob}, increasing along the vehicle path at each of the fingerprint poses. Solid red lines represent the vehicle's global uncertainties at the SIFT fingerprint places. Numbers represent each fingerprint. The graph also shows an example of a shorter path selection for global uncertainty calculation after a loop-closing situation.

5.1. Local sub-maps

Each time a new fingerprint is taken, an associated sub-map is created. The vehicle's relative local pose $X^{fp_l}_{\text{rob}}$ and its covariance $P^{fp_l}_{\text{rob}}$ are stored in the fingerprint at that moment.

Due to the need of being aware about the current global uncertainty at any time, we need to maintain P^G_{rob} updated (see Fig. 4). We calculate it by using the *coupling summation formula* (see [23]) in a recursive way. The process can be summarized as follows: first, to obtain P^G_{rob} we need to solve (3).

$$P^G_{\text{rob}} = \frac{\partial X^0_{\text{rob}}}{\partial X^0_{fp_l}} \cdot P^0_{fp_l} \cdot \left(\frac{\partial X^0_{\text{rob}}}{\partial X^0_{fp_l}}\right)^T + \frac{\partial X^0_{\text{rob}}}{\partial X^{fp_l}_{\text{rob}}} \cdot P^{fp_l}_{\text{rob}} \cdot \left(\frac{\partial X^0_{\text{rob}}}{\partial X^{fp_l}_{\text{rob}}}\right)^T. \tag{3}$$

$X^{fp_l}_{\text{rob}}$ expresses the local vehicle's pose relative to the current fingerprint and X^0_{rob} and $X^0_{fp_l}$ expresses the vehicle and current fingerprint absolute poses respectively.

Second, to obtain the global covariance of the current fingerprint $P^0_{fp_l}$, we must apply (3) again, but this time to the previous fingerprint, as shown in (4).

$$P^0_{fp_l} = \frac{\partial X^0_{fp_l}}{\partial X^0_{fp_{l-1}}} \cdot P^0_{fp_{l-1}} \cdot \left(\frac{\partial X^0_{fp_l}}{\partial X^0_{fp_{l-1}}}\right)^T + \frac{\partial X^0_{fp_l}}{\partial X^{fp_{l-1}}_{fp_l}} \cdot P^{fp_{l-1}}_{fp_l} \cdot \left(\frac{\partial X^0_{fp_l}}{\partial X^{fp_{l-1}}_{fp_l}}\right)^T. \tag{4}$$

We apply the same iterative procedure until we reach the first fingerprint, where $P^0_{fp_l} = P^{fp_0}_{fp_l}$ can be directly solved.

At the time of sub-map creation, the current vehicle's local uncertainty $P^{fp_{l+1}}_{\text{rob}}$, conditioned to the new sub-map and on its own frame, is set to 0 at the beginning. So, we assume a certain position of the vehicle with respect to the newly created sub-map. The current visible landmarks were observed within the previous sub-map fp_l, however, we remove them from that sub-map and incorporate them in the new one fp_{l+1}. Therefore, we start the new sub-map with a number of already initialized landmarks, which will have new local coordinates $Y^{fp_{l+1}}_i$ expressed on the new sub-map. So, the total sub-map state vector starts in the following form $X^{fp_{l+1}} = \left(X^{fp_{l+1}}_{\text{rob}} \quad Y^{fp_{l+1}}_1 \quad Y^{fp_{l+1}}_2 \quad \cdots\right)^T$. To calculate $Y^{fp_{l+1}}_i$ from their expression on the previous sub-map $Y^{fp_l}_i$, we apply the *common root coupling* formula proposed on [23]. It allows changing the base reference from fp_l to fp_{l+1} using the common reference of the vehicle $X^{fp_{l+1}}_{\text{rob}}$ on the new sub-map. We define $X^{fp_{l+1}}_{\text{rob}} = 0$ because it is the base reference of fp_{l+1} at that time.

To obtain the landmark's covariances expressed on the fp_{l+1} base frame we make use of the common root coupling as well (5). If we assume $P^{fp_{l+1}}_{\text{rob}} = 0$, the second term of Eq. (5) disappears and $P^{fp_{l+1}}_{Y_iY_i}$ depends only on $P^{\text{rob}}_{Y_iY_i}$, which represents the uncertainty of the landmark's positions in the vehicle base frame.

$$P^{fp_{l+1}}_{Y_iY_i} = \frac{\partial Y^{fp_{l+1}}_i}{\partial Y^{\text{rob}}_i} \cdot P^{\text{rob}}_{Y_iY_i} \cdot \left(\frac{\partial Y^{fp_{l+1}}_i}{\partial Y^{\text{rob}}_i}\right)^T + \frac{\partial Y^{fp_{l+1}}_i}{\partial X^{\text{rob}}_{fp_{l+1}}} \cdot P^{\text{rob}}_{fp_{l+1}} \cdot \left(\frac{\partial Y^{fp_{l+1}}_i}{\partial X^{\text{rob}}_{fp_{l+1}}}\right)^T. \tag{5}$$

In contrast to our method, in [24] they share landmarks between sub-maps, in cases where the number exceeds a threshold. Then, they create a link between the two sub-maps, expressed through a similarity transformation. This way, measuring shared landmarks allow not only the optimization of the local sub-map but also the global one, even without closing large loops. Because of the large size of the outdoor environments, which is the objective of our work, common places, belonging to different sub-maps, are not often visible at the same time, and therefore no inter-node links will be usually added. Also data association on low level landmarks is a quite difficult task on this kind of large environment. Therefore, we solve that issue defining an especial kind of fingerprint, denoted SIFT fingerprints, which identify singular places, being able to re-identify them, closing a loop and optimizing the global map. This is explained in the next sub-sections.

5.2. SIFT fingerprints

Our system identifies a specific place using the SIFT fingerprints. These fingerprints, apart from the vehicle's pose, are composed

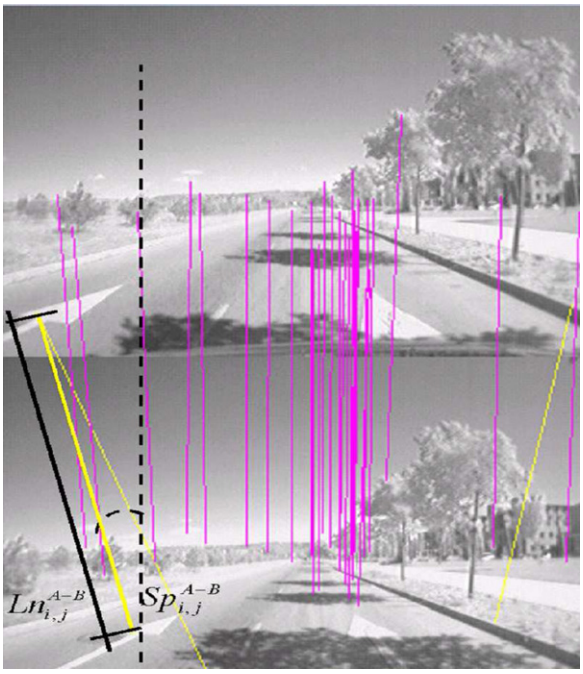

Fig. 5. Fingerprint SIFT features matching. Outliers are marked in light colors. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of a number of SIFT landmarks distributed across the reference image and characterize the visual appearance of the image. SIFT features were introduced by Lowe in [32–34]. SIFT features are invariant to image scaling and rotation, and partially invariant to changes in illumination and the 3D camera's viewpoint. In addition, the features are highly distinctive, which allows a single feature to be correctly matched with a high probability. This is achieved by the association of a 128 length descriptor to each of the features, which will identify uniquely all of them. These SIFT feature descriptors $\vec{\delta}$ are loaded in each SIFT fingerprint joint to the left image coordinates and the 3D vehicle's position $Yf_m^q = \left(u_L \quad v_L \quad X \quad Y \quad Z \quad \vec{\delta}\right)$ for the fingerprint matching process.

5.3. Loop closing detection

One of the main issues concerning SLAM in large environments is the *loop-closing* problem. The first issue to solve is the recognition of previously visited places. Once a new SIFT fingerprint is generated it is matched with all stored SIFT fingerprints within the uncertainty area defined by P_{rob}^G. This matching is carried out for each pair of SIFT fingerprints (sf_A, sf_B), taking into account both the number of recognized SIFT features and their relative positions within the images to compare. The overall process is as follows:

1. Computation of the Euclidean distance between the descriptors $\vec{\delta}_i^A$, $\vec{\delta}_j^B$ of all detected SIFT features on both fingerprints (sf_A, sf_B), which is shown in (6).

$$\left\{ \left\| \vec{\delta}_1^A - \vec{\delta}_1^B \right\|, \ldots, \left\| \vec{\delta}_1^A - \vec{\delta}_{m_B}^B \right\|, \ldots, \left\| \vec{\delta}_{m_A}^A - \vec{\delta}_{m_B}^B \right\| \right\}. \tag{6}$$

Then, we select those close enough as correctly matched. The trigger value is empirically selected.

2. Lines connecting each pair of matched features are calculated. The corresponding lengths $Ln_{i,j}^{A-B}$ and slopes $Sp_{i,j}^{A-B}$ are computed as well as we depict on Fig. 5.

3. Outlier features are excluded from the computation by using the *RANSAC* method. The model to fit is defined as the vector $\left(\text{avg}(Ln_{i,j}^{A-B}), \text{avg}(Sp_{i,j}^{A-B})\right)$, containing the average lengths and slopes of the connecting lines. RANSAC is applied to the whole set of lines, calculating the Euclidean distance of all the individual length/slope pairs to the average. Features whose connecting lines pairs are close enough to the model are considered as inliers, otherwise they are declared as outliers.

4. The global *fingerprint matching probability* is computed as a weighted function of 2 parameters: *Number of matched features probability* $P(n_{mt}) = n_{mt}/m_3$ and Inliers/n_{mt} *relation*, where n_{mt} represents the total number of matches (inliers + outliers) and (m_1, m_2, m_3) were experimentally obtained:

$$P_{fp\_\text{match}} = m_1 \cdot P\left(n_{mt}\right) + m_2 \left(n_I/n_{mt}\right). \tag{7}$$

Obviously, $P(n_{mt})$ can eventually be higher than 1, so we limited the function to avoid this situation. Typical values for our experiments are $m_1 = 2/3$, $m_2 = 1/3$ and $m_3 = 40$.

5.4. Map correction

Once a loop-closing has been detected, the whole map must be corrected according to the old place recognized. To do that, we use the MLR algorithm [17]. The purpose of this algorithm is to assign a globally consistent set of Cartesian coordinates to the fingerprints of the graph based on local, inconsistent measurements, by trying to maximize the total likelihood of all measurements. The reasons for using it have been its highly efficient implementation in terms of computational cost and the extremely high complexity allowed for the relations between new and previously visited places.

This algorithm provides the ability of closing multiple loops even in a hierarchical way. The MLR inputs are the relative poses and covariances of the fingerprints. As outputs MLR returns the most "likely" set of fingerprint poses, i.e., the set already corrected. Due to the standard MLR does not provide corrected covariances we have modified the method to calculate them.

The MLR algorithm manages only 2D information, therefore we need to obtain the 2D related fingerprint pose $X_{2D}^{fp_l}$ and covariance $P_{2D}^{fp_l}$ from $X_{fp_l}^{fp_{l-1}}$ and $P_{fp_l}^{fp_{l-1}}$. First, the 2D pose is defined as: $X_{2D}^{fp_l} = \left(x_{2D} \quad y_{2D} \quad \theta_{2D}\right)^T$, i.e., the 2 planar coordinates and the orientation angle. Then, we can relate both 2D and 3D poses as shown in (8).

$$X_{2D}^{fp_l} = \left(x_{fp_l}^{fp_{l-1}} \quad z_{fp_l}^{fp_{l-1}} \quad 2\arccos\left(q0_{fp_l}^{fp_{l-1}}\right)\right)^T. \tag{8}$$

Also, we compute the 2D covariance by using the corresponding Jacobians depicted in (9).

$$P_{2D}^{fp_l} = \frac{\partial X_{2D}^{fp_l}}{\partial X_{fp_l}^{fp_{l-1}}} \cdot P_{fp_l}^{fp_{l-1}} \cdot \left(\frac{\partial X_{2D}^{fp_l}}{\partial X_{fp_l}^{fp_{l-1}}}\right)^T. \tag{9}$$

The MLR algorithm is based on a simpler one, which is called SLR (*Single Level Relaxation*). The basic steps of the SLR are, first compute a quadratic error function of the fingerprints with the form:

$$\Psi^2\left(X_M\right) = \left(X_M\right)^T A_M X_M - 2\left(X_M\right)^T b_M \tag{10}$$

where $X_M = \left(Xc_{fp_1}^0 \quad Xc_{fp_2}^0 \quad \cdots \quad Xc_{fp_L}^0\right)^T$ represents the total vector of the whole set of 2D corrected poses. In this case, the poses are expressed in global coordinates. After that, it finds X_M to minimize Ψ^2, which is done by solving $A_M X_M = b_M$. The efficient way of solving this equation is the key of the relaxation technique.

The Ψ^2 error function is defined specifically as follows:

$$\Psi^2\left(X_M\right) = \sum_{l=0}^{L} \left(\eta_{2D}^{fp_l}\right)^T \left(P_{2D}^{fp_l}\right)^{-1} \eta_{2D}^{fp_l} \tag{11}$$

where, for each of the fingerprints:

$$\eta_{2D}^{fp_l} = f_M\left(Xc_{fp_l}^0, Xc_{fp_{l-1}}^0\right) - X_{2D}^{fp_l} \tag{12}$$

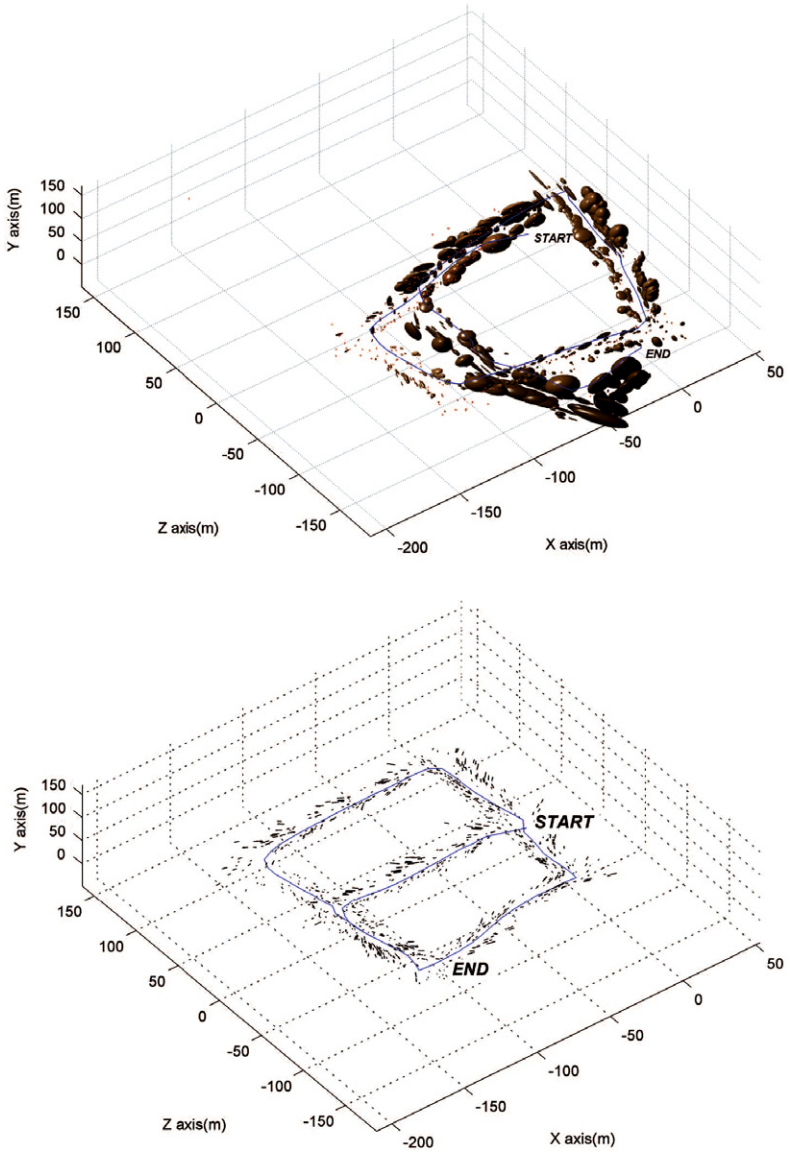

Fig. 6. Detail on the first test path showing landmark global uncertainties as a result of only the low level SLAM estimation (up) and after applying high level SLAM optimization (down).

f_M expresses $Xc^0_{fp_l}$ into $Xc^0_{fp_{l-1}}$ coordinates, therefore $\eta^{fp_l}_{2D}$ is the difference between the corrected pose and the estimated one. Linearizing f_M as shown in [17], Eq. (11) can be expressed in the form of (10). The basic idea of the relaxation is to exploit the sparsity of $A_M X_M = b_M$ and solve it one block-row at a time, corresponding to one of the fingerprint poses $Xc^0_{fp_l}$ of the total vector X_M, considering all the rest as constant. Repeating the procedure for the rest of fingerprints in an iterative procedure the equation is efficiently solved.

The MLR improvement is based on the idea of simplifying the calculation of $A_M X_M = b_M$ by reducing (discretizing) the number of poses iteratively to a half each time. Several hierarchical levels are defined, one per each discretization step. At the coarsest level, the residual equation is directly solved using the Cholesky factorization method. Finally, the solution is interpolated through each of the levels to the finest one in order to obtain the result of the original equation. This down–up–down cycle is known as the V-cycle.

Once the 2D corrected vector has been calculated we obtain the corresponding 3D corrected fingerprints. At the step of obtaining 2D from the 3D poses (see (8)), we lose the $y^{fp_{l-1}}_{fp_l}$ coordinate (altitude) information. Therefore, when going back from 2D to 3D again we have to set this value. We take the assumption of a flat terrain, because our system is mounted on a commercial car driving in a flat urban area, so, this value will be taken as 0. Then, we form the corrected pose vector for each fingerprint as:

$$X^0_{fp_l} = \left(x^0_{fp_l} \quad 0 \quad y^0_{fp_l} \quad \cos\frac{\theta^0_{fp_l}}{2} \quad 0 \quad \sin\frac{\theta^0_{fp_l}}{2} \quad 0 \right)^T. \tag{13}$$

As we explained before, the standard MLR method does not provide a means to obtain the corrected global covariances of the fingerprints. The reason is because the method is based uniquely on the relative covariances between poses. As we have shown, our system does not need to know the global covariances to perform a map optimization. However, in order to have a rough estimation of the revisited SIFT fingerprints, it is needed to keep the global uncertainty of the vehicle updated. After we close a loop, there is a situation where one fingerprint has relations with more than one additional fingerprint, as occurs, for example, to sf_3 (see Fig. 4). To calculate the global vehicle uncertainty P^G_{rob}, we must apply the recursive coupling formula showed in (3). In order to reach *rob* position we can couple local fingerprints uncertainties starting

from fp_0 going through the shaded node's path or also covering the white node's path instead. Due to the shorter path, choosing the first option will lead to a lower P^G_{rob} than choosing the second. By closing the marked loop we have implicitly reinitialized the global uncertainty at that moment to the one associated to sf_3, therefore reducing it. So as a rule, to calculate the current P^G_{rob} we apply the recursive formula to the shortest possible path from the first fingerprint to the current position.

Being aware of the current global uncertainty is important in order to increase the fingerprint search process efficiency because the number of SIFT fingerprints matched will be lower.

The last step is to transfer the correction performed on the high SLAM level into the Low SLAM level. This is done by applying the same transformation of each fingerprint pose to all the landmarks within the sub-map. By doing this, we keep the relative positions of the landmarks unchanged with respect to their corresponding local sub-map reference frame. Therefore, landmark covariances remain unchanged in the frame of each sub-map. However, to represent their global uncertainties, we show on Fig. 6 a portion of one of the paths used for testing purposes. We represent the global feature covariances using just the EKF on the local maps (Fig. 6 (up)) and after applying the MLR optimization (Fig. 6 (down)).

6. Results

Although the system has been designed to work online and several online tests were carried out, to further analyze its behavior several video sequences were collected from a commercial car manually driven in large urban areas. The employed cameras for the stereo pair were the Unibrain Fire-i IEEE1394 with additional wide-angle lens, which provide a field of view of around 100° horizontal and vertical with a resolution of 320 × 240.

The baseline of the stereo camera was 40 cm. Both cameras were synchronized at the time of commanding the start of transmission. The cameras were mounted inside the car on the top of the windscreen. The calibration was performed offline using a chessboard panel using the method referenced in [32]. The first video sequence was taken by covering with the car the urban path showed on Fig. 7. The average speed of the car was around 30 km/h. The complete covered path was 2.27 km long. It contained 3 loops inside, taking 7250 low level landmarks and 235 fingerprints.

The second one was taken on an urban environment as well, and followed the path showed on Fig. 8. The average speed of the car was approximately the same than in the first sequence, but in this case the length covered was 2.19 km. It contained 4 loops inside, taking 8130 low level landmarks and 230 fingerprints. To evaluate the performance of our system we compared our results with a ground truth reference. This ground truth was obtained with an RTK-GPS Maxor GGDT, which provides an estimated accuracy of 2 cm. On the other hand, we collected together car positions obtained from a standard low-cost GPS Navilock NL-302U with an accuracy ranging from 1.5 m to 6 m, to analyze them, taking in mind a future integration of this sensor in our current SLAM system. Fig. 9 depicts the estimation of our SLAM system and the standard GPS compared to the ground truth. We can highlight the relatively low error on the initial part (up) of the SLAM estimation, taking into account that no other sensors were used to help on that task. Because of the long length of the straight segment going to the lower part (about 350 m) and the reduced number of near landmarks taken in this section due to the open-spaced environment (no buildings close the path), there is a significant accumulated error on the estimation of the trajectory.

However, it has to be noticed that the relative error, once closed that loop, is still low.

We have also calculated the mean error relative to the ground truth of both the standard GPS and our SLAM implementation (see Fig. 10). The first observation is that the error using our method is around 60 m as much. Obviously, as long as the vehicle covers more distance the errors using only visual SLAM are higher than using a GPS. However, looking at the GPS error, we observe that, at certain parts of the path, this error is very high because no GPS data were received. This effect was due to the high buildings located in that place, which caused the satellite signals to be not visible and as a consequence the GPS was unable to provide a location of the vehicle. As a greater number and taller buildings are in an urban area a greater probability of GPS loss exists. Therefore, in these cases, even the absolute error on the estimation provided by our system is much lower than the GPS one. Moreover, focusing in the moment right after the second period of GPS loss, we can see how the error on the GPS estimation grows clearly faster than our method. This means that, at this time the visual data is more reliable than the GPS. Taking into account the results, if we combine both visual SLAM and a low cost GPS, we should be able to obtain a similar accuracy of a high quality GPS at a much lower cost. Therefore, as a future work, we plan to integrate both the standard GPS and visual sensors to improve the global estimation. On Fig. 11 we show the map representation for the estimation made by our system. In this case we applied the system to the second test environment. The low level landmarks are marked in a yellow color. The ordinary fingerprints, drawn in a green color, are also marked with their associated identification number. On each of the turns performed by the vehicle, a SIFT fingerprint was taken. These fingerprints are shown in a red color. Some of these places were the ones where the 4 loops closings took place.

With respect to the processing time, the real-time implementation imposes a time constraint, which shall not exceed 33 ms for a 30 frames per second capture rate.

All results were taken using an AMD Turion 2.0 GHz CPU. Fig. 12 depicts the total processing times along the whole vehicle path for the first test. As we can see our method is able to work under a real time constraint, the average processing time remaining quite

Table 1
Processing times.

Low level SLAM processing times		High level SLAM processing times (parallelized)	
Number of features / frame	5	Number of features	7250
		Number of fingerprints	235
Filter step	Time		Time
Measurements	3 ms	Fingerprint matches	3 s
Filter update	5 ms	Loop closing	1 s
Feature initializations	7 ms		

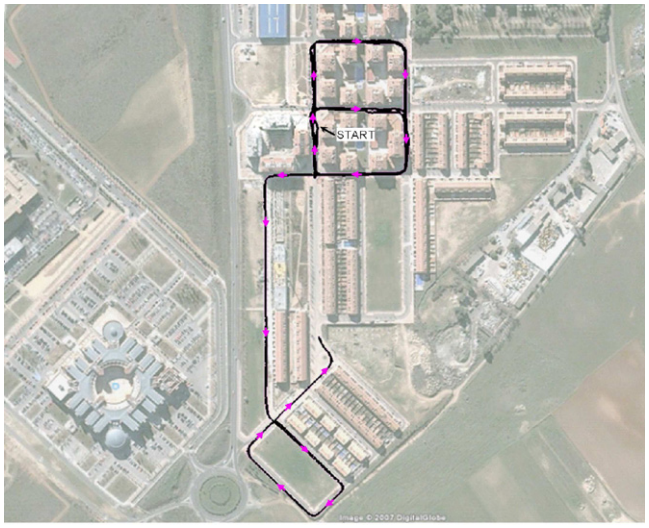

Fig. 7. Aerial view of the path for the first test. The starting point is indicated.

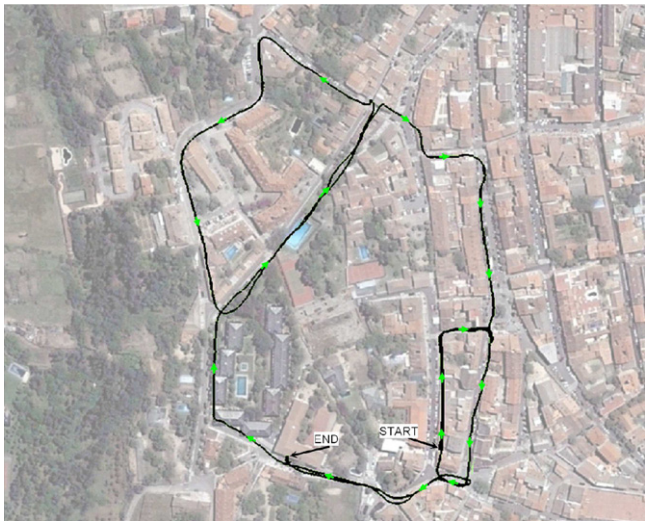

Fig. 8. Aerial view of the path for the second test case. The starting and end point is indicated.

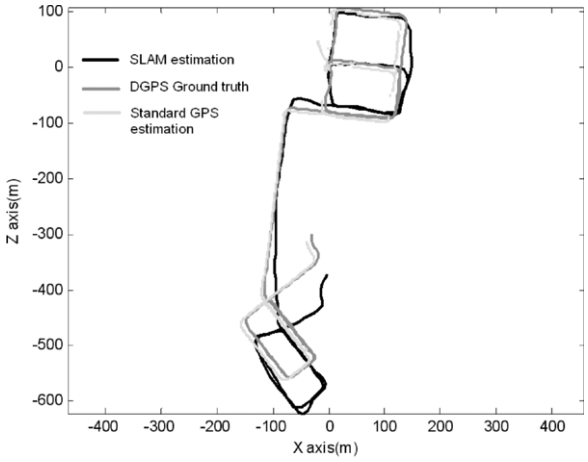

Fig. 9. Path estimation using our SLAM method, a standard low-cost GPS and the ground truth.

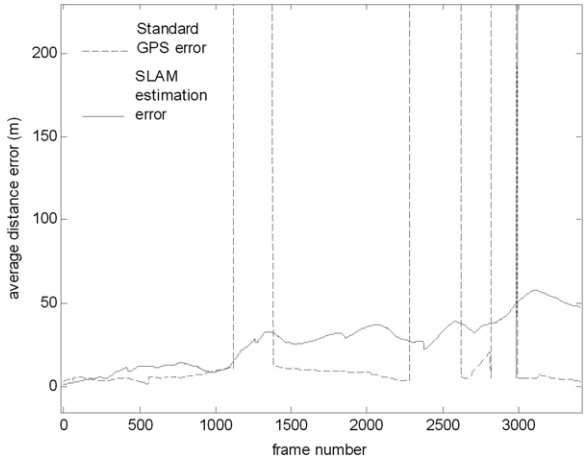

Fig. 10. Average distance error on the path using standard GPS (dashed line) and our SLAM system (solid line) relative to the RTK-GPS reference.

constant along the whole path, even during loop closing situations. On Table 1 we show the average processing times for some of the most important tasks in the process. Focusing on the low level SLAM tasks, we can see that a higher time is used on the landmark's

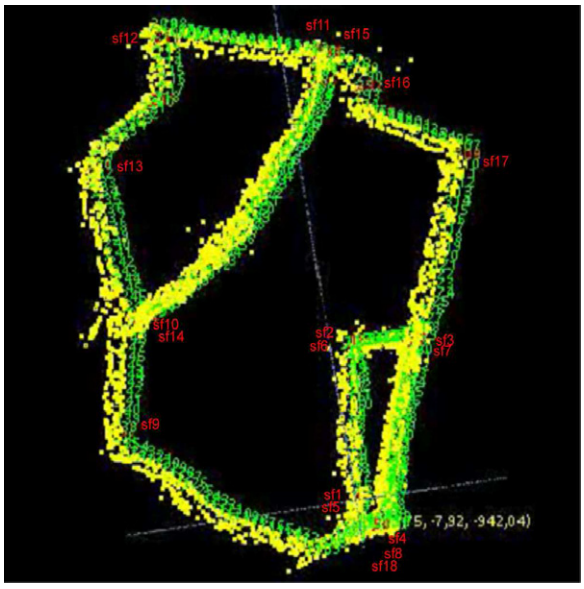

Fig. 11. Map representation estimated by our system. In yellow we depict the visual landmarks acquired by the system. The ordinary fingerprints are shown and numbered in a green color. The SIFT fingerprints are represented by a red color (on the turns of the vehicle). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

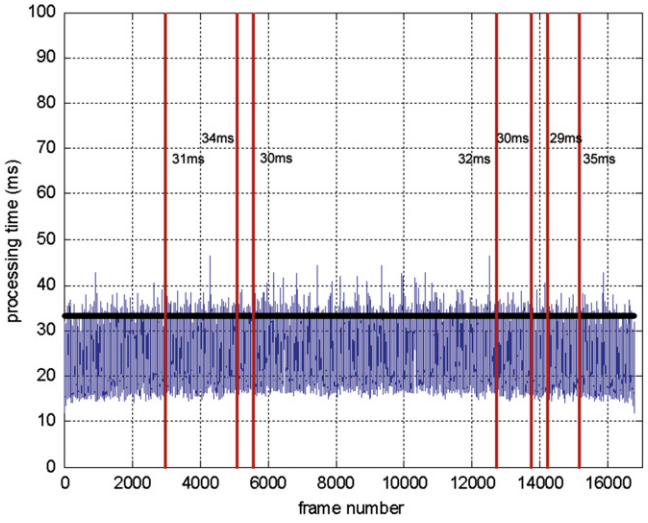

Fig. 12. Processing times for the whole tasks. Real time limit is represented as a constant 33 ms black line. Frames where a loop closing takes place are marked using vertical lines, showing the global processing time value associated to them. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

initialization phase due to the large search area along the epipolar line, even though we restricted its length for 1 m $\rightarrow \infty$ search range.

Regarding the high level SLAM, time dedicated to *SIFT fingerprint matching* process as well as the correction of the map at the time of loop closing, having 7250 landmarks, is slightly higher than real time. However, both tasks do not belong to the continuous self-locating process carried out by the low level SLAM, so, there is no need to complete them within a single frame time slot. Therefore, we can obtain a positive fingerprint matching result some few frames after it was really detected. Then, we can go back and start the loop-closing task. This implies that both of these tasks can be computed in *parallel*, keeping them outside the real time computation. A similar idea was recently presented in [35]. They implemented a SLAM system for a small indoor workspace,

Table 2
Performance and test results of different SLAM algorithms.

Method	Memory	Update	Global update	Loop closing	Measured update	Measured loop
Maximum likelihood	m	$(n+p)^3$	$(n+p)^3$	$(n+p)^3$	≫EKF	≫EKF
EKF	n^2	n^2	n^2	n^2	≫CEKF	≫CEKF
CEKF	$n^{3/2}$	k^2	$kn^{3/2}$	$kn^{3/2}$	232 ms	82.2 s
SLR	kn	kn	kn	kn^2	24 ms	4.2 s
FastSLAM	Mn	$M \log n$	$M \log n$	$M \log n$	339 ms	339 ms
SEIF	kn	k^2	k^2	k^2	–	–
TJTF	k^2n	k^3	k^3n	k^3n	–	–
Treemap	kn	k^2	$k^3 \log n$	$k^3 \log n$	22 ms	966 ms
MLR	kn	kn	kn	kn	935 ms	935 ms
Hierarchical: MLR + EKF	kn	k^2	kn	kn	21 ms	935 ms

Table 3
Robustness to illumination changes.

% False positives / % False negatives	Daylight morning	Daylight afternoon	At sunset	At night
Daylight morning	0/7.5	1/10	0/10	0/40
Daylight afternoon		1/7.5	0/12.5	0/32.5
At sunset			0/10	0/35
At night				0/20

where tracking and location tasks, in one hand and mapping and optimization tasks in the other hand are split independently. Then, both of them are computed in parallel using a dual-core processor. The main difference with our approach is that we maintain a joint location and mapping low-level task, while adding an additional higher level global optimization process, which is computed in parallel. We apply our method to large-scale outdoor environments. So, there is not much advantage on implementing a separated mapping process due to the need of including new landmarks continuously. Also, we keep the ability of continuously optimize the local maps thanks to the joint low level SLAM process.

On Table 2 we compare the memory requirements and the computational cost of our system with respect to other well-known methods, according to the operation number carried out for each stage of the algorithm. We have based this study on the figures presented in [17]. In the table, n is the total number of landmarks of the global map, m measurements, p robot poses and k landmarks within the local sub-map. We have tested most of these methods, obtaining the average computation times showed on the table. Loop detection + global map optimization times are obtained without the implementation of any concurrent method. As can be observed, the lowest time consuming method is the MLR applied to our hierarchical SLAM implementation.

On Table 3 we show a comparative study of the robustness to illumination changes. We focused on the SIFT fingerprint matching process. We took a 40 image database of the same place at different times along the day. We registered the number of erroneous matchings (false positives) as well as missing ones (false negatives). From the results we can conclude that the probability of a false positive is extremely low, keeping reasonable values for false negatives in daylight. During the night results get worse on false negatives, mainly due to the decrease of illuminated areas.

7. Conclusion

In this paper we have presented a hierarchical SLAM of two levels (topological/metric) that allows self-locating a vehicle in a large-scale outdoor urban environment using a wide-angle stereo camera as the only sensor. Using this hierarchical strategy, on one hand, we keep a local consistency of the metric sub-maps by mean of the EKF (low SLAM level) and global consistency by using a topological map and the MLR algorithm. On the other hand, our method is able to work under a real time constraint, the average processing time remaining quite constant for very large-scale environments. We have shown that our visual SLAM can improve the accuracy of a low-cost GPS under certain circumstances, enhancing its behavior. Therefore combining both low-cost GPS and vision we can reach a similar accuracy to a high quality GPS. One limitation of our system is that a flat terrain is assumed for matching the 2D map of the topological level with the 3D maps of the metric one. Our method can cope with 3D motions to a certain extent but a graceful degradation in map accuracy appears as the roughness of the terrain increases. In extreme cases it is possible that our method would create inconsistent maps. On the other hand, loop closing detection strongly depends on the visual appearance of images taken almost in the same place. As future work, we plan to generalize the MLR algorithm in order to manage 3D characteristics and to fuse a low-cost GPS sensor with our current system to improve the loop closing detection and the GPS losses. Our final goal is the autonomous outdoor navigation of a vehicle in large-scale urban environments. Regarding processing times, the multi-hypothesis tracking application will be studied.

References

[1] M.E. López, L.M. Bergasa, R. Barea, M.S. Escudero, A navigation system for assistant robots using visually augmented pomdps, Autonomous Robots 19 (1) (2005) 67–87.
[2] P. Newman, J. Leonard, J. Tardós, J. Neira, Explore and return: experimental validation of real time concurrent mapping and localization, in: Proc. IEEE Int. Conf. Robotics and Automation, IEEE, 2002, pp. 1802–1809.
[3] A. Davison, Real-time simultaneous localisation and mapping with a single camera, in: Proc. of the 9th International Conference on Computer Vision ICCV'03, 2003.
[4] C. Engels, H. Stewenius, D. Nister, Bundle adjustment rules, in: PCV06, 2006.
[5] E. Royer, M. Lhuillier, M. Dhome, T. Chateau, Localization in urban environments: monocular vision compared to a differential GPS sensor, in: CVPR'05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'05—Volume 2, IEEE Computer Society, Washington, DC, USA, 2005, pp. 114–121.
[6] E. Mouragnon, F. Dekeyser, P. Sayd, M. Lhuillier, M. Dhome, Real time localization and 3D reconstruction, in: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1, 2006, pp. 363–370.
[7] M. Montemerlo, Fastslam: a factored solution to the simultaneous localization and mapping problem with unknown data association, Ph.D. Thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003.
[8] T. Bailey, J. Nieto, E.M. Nebot, Consistency of the fastslam algorithm, in: ICRA, IEEE, 2006, pp. 424–429.
[9] C. Stachniss, G. Grisetti, W. Burgard, Analyzing Gaussian proposal distributions for mapping with rao-blackwellized particle filters, San Diego, CA, USA, 2007.
[10] E. Nerurkar, S. Roumeliotis, Power-slam: a linear-complexity, consistent algorithm for slam, in: Intelligent Robots and Systems, 2007, IROS 2007, IEEE/RSJ International Conference on, 2007, pp. 636–643.
[11] F. Dellaert, M. Kaess, Square root sam: simultaneous localization and mapping via square root information smoothing, International Journal of Robotics Research 25 (12) (2006) 1181–1203.

[12] L.A. Clemente, A. Davison, I. Reid, J. Neira, J.D. Tardós, Mapping large loops with a single hand-held camera, in: W. Burgard, O. Brock, C. Stachniss (Eds.), Robotics: Science and Systems, The MIT Press, 2007.
[13] J. Tardós, J. Neira, P. Newman, J. Leonard, Robust mapping and localization in indoor environments using sonar data, The International Journal of Robotics Research 21 (4) (2002).
[14] L.M. Paz, P. Piniés, J.D. Tardós, J. Neira, 6 DoF slam with stereo-in-hand, in: IEEE International Conference on Robotics and Automation, 2008.
[15] M. Bosse, P. Newman, J. Leonard, S. Teller, Simultaneous localization and map building in large-scale cyclic environments using the atlas framework, The International Journal of Robotics Research 23 (12) (2004) 1113–1139.
[16] H. Chang, C. Lee, Y. Hu, Y.-H. Lu, Multi-robot slam with topological/metric maps, in: Intelligent Robots and Systems, 2007, IROS 2007, IEEE/RSJ International Conference on, 2007, pp. 1467–1472.
[17] U. Frese, P. Larsson, T. Duckett, A multilevel relaxation algorithm for simultaneous localization and mapping, IEEE Transactions on Robotics and Automation 21 (2) (2005) 196–207.
[18] A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, Monoslam: real-time single camera slam, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (6) (2007) 1052–1067.
[19] P. Piniés, J. Tardós, Scalable slam building conditionally independent local maps, in: Intelligent Robots and Systems, 2007, IROS 2007, IEEE/RSJ International Conference on, 2007, pp. 3466–3471.
[20] J. Leonard, H. Feder, Decoupled stochastic mapping, in: Technical Report, MIT, 1999.
[21] K. Chong, L. Kleeman, Large scale sonarray mapping using multiple connected local maps, in: International Conference on Field and Service Robotics, 1997, pp. 538–545.
[22] S. Williams, Efficient solutions to autonomous mapping and navigation problems, Ph.D. Thesis, University of Sydney, 2001.
[23] T. Bailey, Mobile robot localisation and mapping in extensive outdoor environments, Ph.D. Thesis, University of Sydney, 2002.
[24] E. Eade, T. Drummond, Monocular slam as a graph of coalesced observations, in: Computer Vision, 2007, ICCV 2007, IEEE 11th International Conference on, 2007, pp. 1–8.
[25] C. Estrada, J. Neira, J.D. Tardós, Hierarchical slam: real-time accurate mapping of large environments, IEEE Transactions on Robotics 21 (4) (2005) 588–596.
[26] B. Steder, G. Grisetti, S. Grzonka, C. Stachniss, A. Rottmann, W. Burgard, Learning maps in 3D using attitude and noisy vision sensors, in: Iros, San Diego, CA, USA, 2007.
[27] H. Andreasson, T. Duckett, A. Lilienthal, Mini-slam: minimalistic visual slam in large-scale environments based on a new interpretation of image similarity, in: Robotics and Automation, 2007, IEEE International Conference on, 2007, pp. 4096–4101.
[28] F. Fraundorfer, C. Engels, D. Nister, Topological mapping, localization and navigation using image collections, in: Intelligent Robots and Systems, 2007, IROS 2007, IEEE/RSJ International Conference on, 2007, pp. 3872–3877.
[29] M. Cummins, P. Newman, Probabilistic appearance based navigation and loop closing, in: Robotics and Automation, 2007 IEEE International Conference on, 2007, pp. 2042–2048.
[30] D. Schleicher, L. Bergasa, R. Barea, E. Lopez, M. Ocana, Real-time simultaneous localization and mapping using a wide-angle stereo camera and adaptive patches, in: Intelligent Robots and Systems, 2006, IEEE/RSJ International Conference on, 2006, pp. 2090–2095.
[31] D. Schleicher, L. Bergasa, R. Barea, E. Lopez, M. Ocana, J. Nuevo, Real-time wide-angle stereo visual slam on large environments using sift features correction, in: Intelligent Robots and Systems, 2007, IROS 2007, IEEE/RSJ International Conference on, 2007, pp. 3878–3883.
[32] D. Lowe, Object recognition from local scale-invariant features, in: Computer Vision, 1999, The Proceedings of the Seventh IEEE International Conference on, vol. 2, 1999, pp. 1150–1157.
[33] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.
[34] S. Se, D. Lowe, J. Little, Vision-based mobile robot localization and mapping using scale-invariant features, in: Robotics and Automation, 2001. Proceedings 2001, ICRA, IEEE International Conference on, vol. 2, 2001, pp. 2051–2058.
[35] G. Klein, D. Murray, Parallel Tracking and Mapping for Small ar Workspaces, IEEE, ACM, 2007.

David Schleicher received the M.S. degree (First Class Honors) in electronic engineering from the University of Alcalá, Madrid, Spain, in 2002. He is currently working towards the Ph.D. degree at the same university.

His current research interests include computer vision, autonomous vehicles, SLAM in robotics and machine learning.

Luis M. Bergasa (M'04–A'05) received the M.S. degree from the Technical University of Madrid, Madrid, Spain, in 1995, and the Ph.D. degree from the University of Alcalá, Madrid, in 1999, all in electrical engineering. He is currently an Associate Professor at the Department of Electronics, University of Alcalá. His research interests includes real-time computer vision and its applications, particularly in the field of robotics, assistance systems for elderly people, and intelligent transportation systems.

He is the author of more than 60 publications in international journals, book chapters, and conference proceedings.

Dr. Bergasa is a member of the Computer Science Society.

Manuel Ocaña received his Ing. Degree in Electrical Engineering in 2002 from the University of Alcalá, and his Ph.D. degree in Electrical Engineering in 2005 from the University of Alcalá, Alcalá de Henares, Madrid, Spain. From 2002 to 2005 he has been researcher at the Department of Electronics, University of Alcalá, where he is currently an Associate Professor. He has been recipient of the Best Research Award for the 3M Foundation Awards in the category of eSafety in 2003 and 2004. His research interests include robotics localization and navigation, assistant robotics and computer vision and control systems for autonomous and assisted intelligent vehicles. He is the author of more than 20 refereed publications in international journals, book chapters, and conference proceeding.

Rafael Barea received the B.S. degree (First Class Honors) from the University of Alcalá, Madrid, Spain, in 1994, the M.S. degree from the Polytechnic University of Madrid, Madrid, in 1997, and the Ph.D. degree from University of Alcalá in 2001, all in telecommunications engineering. He is currently an Associate Professor in the Electronics Department, University of Alcalá, where has been Lecturer since 1994. His current research interests include bioengineering, medical instrumentation, personal robotic aids, computer vision, system control, and neural networks. He is the author of many refereed publications in international journals, book chapters, and conference proceedings.

Elena López received the B.S. degree in telecommunications engineering in 1994, the M.Sc. degree in electronics engineering in 1999, and the Ph.D. degree in 2004, all from the University of Alcalá, Madrid, Spain. She has been a Lecturer in the Electronics Department, University of Alcalá since 1995. Her current research interests include intelligent control and artificial vision for robotics applications. She is the author/coauthor of numerous publications in international journals and conference proceedings in these research lines.