Robotics and Autonomous Systems 61 (2013) 1539–1558

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Long-term mapping and localization using feature stability histograms

B. Bacca [a,b,*], J. Salvi [b], X. Cufí [b]

[a] Universidad del Valle, Calle 13 No 100-00, A.A. 25360, Cali, Colombia
[b] Universitat de Girona, Campus Montilivi, Building PIV, 17071 Girona, Spain

HIGHLIGHTS

- We proposed a more complete Feature Stability Histogram model.
- The Feature Stability Histogram model is able to be used in current SLAM methods.
- Our approach is able to deal with long-term SLAM runs in dynamic environments.
- Our approach is able to filter out dynamic objects and to reduce the matching effort.
- Our approach is able to update the map in accordance with the changes observed.

ARTICLE INFO

Article history:
Received 25 September 2012
Received in revised form
29 June 2013
Accepted 5 July 2013
Available online 17 July 2013

Keywords:
Long-term localization and mapping
Laser range finder
Particle filters
Omnidirectional vision

ABSTRACT

This work proposes a system for long-term mapping and localization based on the Feature Stability Histogram (FSH) model which is an innovative feature management approach able to cope with changing environments. FSH is built using a voting schema, where re-observed features are promoted; otherwise the feature progressively decreases its corresponding FSH value. FSH is inspired by the human memory model. This model introduces concepts of Short-Term Memory (STM), which retains information long enough to use it, and Long-Term Memory (LTM), which retains information for longer periods of time. If the entries in STM are continuously rehearsed, they become part of LTM. However, this work proposes a change in the pipeline of this model, allowing any feature to be part of STM or LTM depending on the feature strength. FSH stores the stability values of local features, stable features are only used for localization and mapping. Experimental validation of the FSH model was conducted using the FastSLAM framework and a long-term dataset collected during a period of one year at different environmental conditions. The experiments carried out include qualitative and quantitative results such as: filtering out dynamic objects, increasing map accuracy, scalability, and reducing the data association effort in long-term runs.

1. Introduction

For over two decades a solution for the SLAM (Simultaneous Localization and Mapping) problem (2D or 3D) has been the focus of research in mobile robotics. Application fields for SLAM range from service [1], industrial [2], security [3], inspection [4] to space [5] sectors. However, the aforementioned application fields also require an autonomous system deployed for long-term operation without human intervention. In these application fields, SLAM is a fundamental task for an autonomous robot since mapping and robot localization are essential to guarantee accurate and safe navigation. Most classic SLAM methods assume a static environment; nevertheless mobile robots require interaction with people and adapt their internal representation of the environment according to the changes taking place in the robot's surroundings.

This work addresses the long-term mapping and localization problem, in which the environment is no longer assumed as static and where many mapping and localization runs can be performed. Nowadays, mobile robots have to deal with dynamic environments, changes in illumination, occlusions by pedestrians, structural changes in the environment, perceptual aliasing, and interaction with people without any previous knowledge of robotics. These environmental changes can be classified as permanent and temporal, and it is desirable that the map representation of the environment changes accordingly. Another important aspect is related to the current number of SLAM methods available (e.g. EKF, appearance-based, pure probabilistic approaches,

* Correspondence to: Universitat de Girona, Campus Montilivi, Building PIV, Office 009, 17071 Girona, Spain. Tel.: +34 972 418976; fax: +34 972 418976.
E-mail addresses: bladimir@eia.udg.edu, bladimir.bacca@correounivalle.edu.co (B. Bacca), qsalvi@eia.udg.edu (J. Salvi), xcuf@eia.udg.edu (X. Cufí).

0921-8890/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.robot.2013.07.003

Table 1
Review of SLAM techniques in dynamic environments.

Authors/year	Classification	Probabilistic framework	Static environment learning	Avoid ever grown number of features	Multiple map representation	Map update
[10] Wang et al./2003	Detect dynamic. obj.	✓	×	×	×	×
[11] Burgard et al./2007	Detect dynamic. obj.	✓	✓	×	✓	✓
[13] Hochdorfer et al./2009	Landmark rating	×	×	✓	×	×
[14] Pirker et al./2011	Landmark rating	×	×	✓	×	×
[12] Andrade-Cetto et al./2002	Landmark rating	×	×	✓	×	×
[16] Konolige and Bowman/2009	Pruning	×	✓	✓	✓	✓
[15] Kretzschmar et al./2010	Pruning	✓	✓	✓	×	✓
[18] Biber and Duckett/2009	Multiple maps	✓	✓	×	✓	✓
[19] Milford and Wyeth/2009	Integrated long-term SLAM	✓	✓	×	×	×
[20] Glover et al./2010	Integrated long-term SLAM	✓	✓	×	×	×
[24] Dayoub et al./2011	Memory management	×	✓	✓	×	✓
[25] Labbe and Michaud/2011	Memory management	×	✓	✓	×	✓

etc.) [6]. Hence, it is necessary to consider the adaptation of all these SLAM methods to operate in long-term mapping and localization conditions.

In this work, the following contributions are made: first, based on our previous work, where the Feature Stability Histogram model was proposed [7], we introduce the probabilistic foundations to integrate the FSH model into the current SLAM solutions. An automatic feature classification method is then proposed to determine whether a feature is stable. Taking advantage of this feature classification, a method to remove useless and weak features caused by dynamic objects is presented. A complete system for long-term mapping and localization is also described, in which the more stable features and environmental configuration at each mapping and localization run are obtained and considered to estimate the robot pose,in this way, these changes in the environment are integrated into the current map model and the internal map representation is updated. Furthermore, the FSH model is analyzed using a metric map built with the FastSLAM [8] approach, which allows us to qualitatively observe the FSH model behavior over the mapping and localization runs in the presence of dynamic objects, and structural changes in the environment. In addition, in this paper we measure the influence of the FSH model over mapping and localization runs in terms of localization accuracy, scalability and matching effort. Finally, successful SLAM methods depend not only on the filter used but on the perception system; therefore, this work describes a sensor model based on the extrinsic calibration between a 2D Laser Range Finder (LRF) and an omnidirectional camera [9,8,7,6,5,4] in order to extract 3D locations of vertical edges, which are used to describe the appearance of the environment.

The remainder of this paper is organized as follows. Section 2 describes related works focused on long-term mapping and localization methods. Subsequently, for self-containment the FSH model is briefly explained in Section 3. The probabilistic foundations used to integrate it with the current SLAM methods are described as well as the feature classification and feature pruning methods in Section 3. The sensor model is then introduced in Section 4. Section 5 presents the experimental conditions and the qualitative and quantitative results. Our final remarks are given in Section 6.

2. Related work

Typical techniques to solve the SLAM problem assume a static environment and many approaches have been proposed in the two last decades. Only in the last eight years has the SLAM problem faced dynamic environments as shown in Table 1.

The most common strategy to deal with dynamic environments is to detect dynamic objects and considering them as spurious measurements. A seminal work using a 3D LRF in outdoor environments was proposed in [10]. Here the Detection and Tracking of Mobile Objects (DTMO) is performed in advance of SLAM. The probabilistic framework proposed in [10] is based on estimating the constant velocity motion of mobile objects and differentiating static from moving observations. In contrast, [11] proposes a probabilistic framework based on occupancy grid maps, estimating whether each individual LRF beam has been reflected by a dynamic object. Once these laser beams are identified, they are filtered out from the range registering process. This technique also learns quasi-static environmental configurations by clustering local grid maps.

A service robot has to learn relevant information about the environment in which it is deployed. One way to do this is through landmark visibility and rating. In this context, [12] developed an EKF-based map building system which incorporates the measurement of landmark strength and quality allowing the elimination of unreliable observations. However, the removal criteria were based on user supplied strength and quality thresholds. In [13] the landmark quality is quantified based on its contribution to the robot's self-localization ability. In these approaches the landmark uncertainty and its visibility are taken into account to quantify the landmark quality, and are then used to avoid the ever-growing number of landmarks. A clustering step is performed to identify the landmark's spatial distribution, which computes their visibility. Another interesting idea is presented in [14], where the landmark visibility is computed using the Histogram of Oriented Cameras (HoC) in order to reduce the average number of matching candidates per frame.

A logical consequence of landmark rating is the pruning of unreliable or useless landmarks. In this sense, [15] is focused on removing observations which do not provide relevant information with respect to the map built so far. To do so, the entropy of an observation is computed using past measurements to obtain its information gain. If this information gain is zero the observation is discarded. In [15] a graph-based SLAM solution is used and the observations are the node positions. In [16] a lifelong mapping solution is proposed using the FrameSLAM approach [17] and a view deletion method based on removing views with low matching rates is described. In spirit this work is similar to [11], but in this case learning clusters of views is done to represent similar and persistent places in the environment. These pruning techniques have the tendency to introduce delayed map updates, since the optimization of large maps can take more time than the observation time window.

One characteristic that has gained attention in long-term mapping and localization is multiple map representations. A seminal contribution on this context was [11,16] with a particularly different approach, however, in [18] the environment is simultaneously represented at multiple timescales and at each timescale a different learning rate is used to obtain a map of the environment. A drawback of [18] is that the timescales were found experimentally, which limits the robot's autonomy when compared to [11,16].

In general, the approaches described so far solve the SLAM problem independently from the detection and tracking of dynamic changes in the environment. Decomposing the whole problem into two separate estimations is understandable given the high dimensionality of the estimation problem, and in many situations the dynamic objects do not provide relevant localization information for the robot. However, in [19] the RatSLAM algorithm is presented and its system architecture is inspired by the rat hippocampus place cells. RatSLAM is composed on an experience space, each experience is associated with a local view cell and each local view cell is associated with a set of pose cells. The map is kept up-to-date by continuously adding experience, and the data association method is strongly dependent on lighting conditions [19]. In contrast, FAB-MAP [20] is a pure probabilistic appearance-based mapping technique with a reliable data association approach, as a result in [21] is presented as a hybrid mapping system combining the best of both algorithms but integrating them in one SLAM solution.

Biologically or psychologically inspired models have been used over many years in the robotics community. The human memory model proposed by [22] or [23], and the recency-weighed averaging approach [18] are good examples of memory management models inspired by psychological and neuroscientific models. In [24] the Atkinson and Shiffrin memory model [22] is considered in order to update the reference view of a particular place. However, the authors assume that the robot is able to self-localize using other means, since their main goal is to keep the reference views of the topological map up to date. Using the human memory model proposed in [23], a real-time loop-closure detection approach is presented in [25], evaluating the number of times locations have been matched and recurrently viewed. This evaluation is done using a Bayesian filter to estimate the probability that the current topological location matches that of one already visited stored in the Working Memory (WM).

The long-term mapping and localization approaches reviewed in this section can integrate (or not) the dynamism of the environment in the estimation process or not. Moreover, given the high number of SLAM solutions available nowadays, it is worth designing and implementing long-term mapping and localization methods in such a way that it can be applied to the current SLAM solutions. In addition, independence in terms of the type of sensor used is another important observation to note from Table 1. Therefore, it seems that a high-level feature management approach such as the FSH model is an interesting option to deal with the long-term mapping and localization problem.

3. Feature stability histogram model

Typical SLAM techniques assume static environments, and they build a map without taking into account real-world conditions, which can include pedestrians, moving obstacles, perceptual aliasing, weather changes, occlusions, and robot–human interaction. So, how can a mobile robot update its location and internal representation of an environment which appearance changes constantly? The Feature Stability Histogram (FSH) is a solution proposed in this work to deal with changing environments and long-term mapping and localization. The main idea behind this is to classify the features of the environment as stable and non-stable, in this way differentiating the most persistent features and environmental configurations from those temporal changes in the environment.

In this section, an overview of the FSH model [7] is presented. The probabilistic foundations are then described in order to integrate the FSH model to the current SLAM methods. And the automatic feature classification and feature pruning methods are also presented. Finally, the SLAM algorithm overview for long-term mapping and localization using the FSH model is described.

3.1. Method overview

For years the scientific community has been finding inspiration in nature, e.g. probabilistic localization has its origins in how the place cells in the hippocampus work. In this paper, the Atkinson and Shiffrin memory model [22] is used as inspiration to distinguish stable features from unstable ones, and to then use the stable features for robot mapping and localization. This model is composed of two main components: the Short-Term Memory (STM), which retains information long enough to use it, and the Long-Term Memory (LTM), which retains information for longer periods of time. If STM inputs are continuously rehearsed, they become part of LTM. The memory model proposed in [22] has drawn criticism from psychologists and neuroscientists due to its extremely linear representation of the memory process [23,26]. They argue that the Atkinson and Shiffrin model does not take into account the ability of many people to recall information despite the fact that this information has not been rehearsed. In addition, this memory model does not consider different levels of memory that could be useful from the robotics point of view, since it can be represented by feature strength.

In this work, a feature management approach for robot mapping and localization inspired by [22] is proposed (see Fig. 1). This modified memory model representation has two main advantages: first, an input feature can bypass STM and become part of LTM depending on the feature strength (e.g., the feature uncertainty, the Hessian value in the SURF descriptor [27] or the matching distance); second, the weighted voting schema implemented to build FSH defines a non-linear memory representation, which means that a feature can be part of STM or LTM depending on its strength. The rehearsal process implemented in this work is based on the number of times a feature was observed. In this way, the appearance of the environment represented for FSH is updated according to the presence or absence of pre-observed features, or the inclusion of new features.

FSH is used to distinguish between STM and LTM features. A feature is defined as an LTM if it has a high value in FSH; otherwise it is considered as an STM feature. This classification has two main advantages: first, it is a straightforward method to deal with dynamic objects because they often produce STM features or spurious features which should not be part of the map until they are rehearsed; and second, it is a suitable method to deal with changing environments. In the end, the more stable features belong to LTM and will be those used for mapping and localization.

The FSH pipeline process is described as follows: the first stage of the FSH model deals with introducing the features set in order to build a normalized histogram using the feature strengths; this is called the Feature Stability Histogram. Afterwards, using the map information the re-observed features can be identified and their votes incremented according with the feature strengths. Finally, as a result of the feature classification as STM or LTM a new model of the environment is obtained as well as a new FSH. It is worth noting that the bigger the FSH value is, the more stable a feature is.

The automatic feature classification process implemented in this work is based on the *k-means* [28] clustering and exponential decay algorithm, such that the feature mean lifetime is identified. Once the LTM features are found, they are used for robot pose estimation and crucial SLAM processes such as loop-closure. In loop-closure situations, the observed LTM features are used to impose constraints in the loop. As a result, when considering LTM features in typical SLAM tasks, the map of the environment can be updated accordingly and the robot can deal with long-term mapping and localization.

3.2. Localization and mapping using the feature stability histogram

The FSH model can be viewed as a method which is transversal to the current SLAM solutions, providing long-term operation and map updating skills. To do so, the SLAM problem formulation and the feature stability level (provided by FSH) are considered

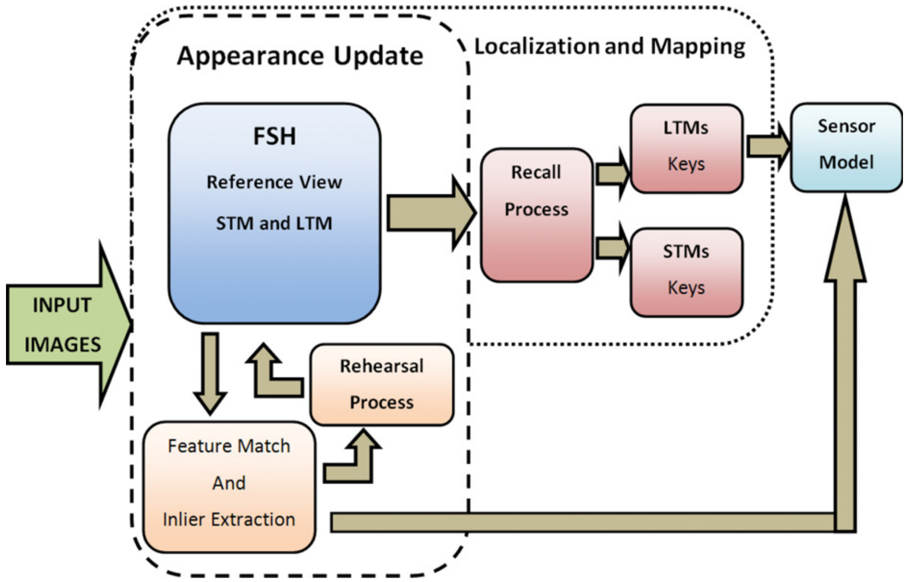

Fig. 1. The modified human memory model.

Table 2
Notation.

Definition	Symbol	Value and uncertainty
Mirror parameter	ζ	0.96651 ± 0.00599
Principal point	u_0, v_0	$510.61716 \pm 1.43172, 420.61415 \pm 1.68635$
Generalized focal lengths	γ_1, γ_2	$402.97909 \pm 0.67177, 403.55496 \pm 0.67395$
Skew	S	0
Distortion parameters	k_1, k_2, k_3, k_4, k_5	$-0.01934 \pm 0.00623, 0.00229 \pm 0.00362, -0.00016 \pm 0.00183, -0.00017 \pm 0.00046, 0$
Translation vector	T	$[-0.0069 \pm 0.0007, -0.2100 \pm 0.0017, 0.5557 \pm 0.027]^{T}$
Roll, Pitch and Yaw	RPY	$0.7644° \pm 0.002°, -4.5928° \pm 0.0005°$ and $-105.0235° \pm 0.033°$

to re-formulate the SLAM problem as shown in Section 3.2.1. The environmental features are classified as LTM or STM in terms of their strength. A detailed description of this process is provided in Section 3.2.2. Also, Section 3.2.2 discusses the ever increasing number of features in long-term operations problem, which is common in service robotics. The integration of the FSH model in a SLAM method involves the modification of the data association, update and loop closure processes, which are described in Section 3.2.3. In order to understand the main variables used in Sections 3.2.1–3.2.3 and their meaning, Table 2 summarizes the notation used.

3.2.1. *Probabilistic foundations*

Fig. 2 shows the modules affected by integrating the FSH model in a typical SLAM framework. Basically, data association, map update and loop-closure processes are involved. The FSH model depends on the data association process to update the histogram of weighted votes. Afterwards, all the re-observed features are estimated, new features are initialized and only the current re-observed LTM features are used to estimate robot position. Furthermore, loop-closure situations require a reliable landmark shared between the current robot position and the re-visited one to close the loop. LTM features in the re-visited position are used to impose reliable constraints and start the preferred non-linear minimization method.

The probabilistic derivation is based on the SLAM problem formulation; however, two important assumptions have to be made: first, the observations can be decomposed into LTM and STM observations. We introduce additional variable $\boldsymbol{C_t}$ which tell us whether an observation belongs to LTM or STM at each time step t. This assumption is depicted in Eq. (1) and shown in Fig. 2:

$$\begin{aligned} \boldsymbol{Z_t} &= \boldsymbol{Z_t^{LTM}} + \boldsymbol{Z_t^{STM}} \\ \boldsymbol{C_t} &= [c_t == LTM,\; c_t == STM] \end{aligned} \tag{1}$$

where $\boldsymbol{Z_t}$ are the observations taken at time t, and $\boldsymbol{C_t}$ is labeled as $c_t = LTM$ or $c_t = STM$ if $\boldsymbol{Z_t}$ corresponds to LTM or STM observations. Second, the posterior of the robot pose depends on the LTM features only, which is shown in Eq. (2):

$$p(\boldsymbol{x_t}, \boldsymbol{M_t}|\boldsymbol{Z_t}, \boldsymbol{U_t}) = p(\boldsymbol{x_t}, \boldsymbol{M_t}|\boldsymbol{Z_t^{LTM}}, \boldsymbol{U_t}) \tag{2}$$

where $\boldsymbol{x_t}$ is the robot state at time t, $\boldsymbol{M_t}$ is the map built so far, $\boldsymbol{U_t}$ are the control inputs at time t, and $\boldsymbol{Z_t^{LTM}}$ are the LTM observations at time t. Next, introducing the variable $\boldsymbol{C_t}$ into the probabilistic formulation of the SLAM problem yields the expression shown in Eq. (3):

$$\begin{aligned} p(\boldsymbol{x_t}, \boldsymbol{M_t}|\boldsymbol{Z_t}, \boldsymbol{U_t}, \boldsymbol{C_t}) &\propto p(\boldsymbol{z_t}, \boldsymbol{c_t}|\boldsymbol{x_t}, \boldsymbol{M_t}, \boldsymbol{Z_{t-1}}, \boldsymbol{U_t}, \boldsymbol{C_{t-1}}) \\ &\quad \times p(\boldsymbol{x_t}, \boldsymbol{M_t}|\boldsymbol{Z_{t-1}}, \boldsymbol{C_{t-1}}, \boldsymbol{U_t}) \end{aligned} \tag{3}$$

where $\boldsymbol{z_t}$ is the last measurement performed. Considering the assumption expressed in Eq. (1), the observation can be factorized out as Eq. (4) shows:

$$\begin{aligned} & p(\boldsymbol{z_t}, c_t|\boldsymbol{x_t}, \boldsymbol{M_t}, \boldsymbol{Z_{t-1}}, \boldsymbol{U_t}, \boldsymbol{C_{t-1}}) \\ &\quad = p(\boldsymbol{z_t}|c_t, \boldsymbol{x_t}, \boldsymbol{M_t}, \boldsymbol{Z_{t-1}}, \boldsymbol{U_t}, \boldsymbol{C_{t-1}}) p(c_t|\boldsymbol{x_t}, \boldsymbol{M_t}, \boldsymbol{Z_{t-1}}, \boldsymbol{U_t}, \boldsymbol{C_{t-1}}) \\ &\quad = p(\boldsymbol{z_t}|c_t, \boldsymbol{x_t}, \boldsymbol{M_t}) p(c_t) \\ &\quad = p(\boldsymbol{z_t}|c_t = LTM, \boldsymbol{x_t}, \boldsymbol{M_t}) p(c_t = LTM) \end{aligned} \tag{4}$$

where the standard Markov assumption was considered, as well as the conditional independence of $\boldsymbol{z_t}$ and c_t given $\boldsymbol{x_t}$ and $\boldsymbol{M_t}$. Furthermore, considering only LTM features (assumption expressed in Eq. (2)), the right term of Eq. (3) can be further developed as depicted in Eq. (5).

$$\begin{aligned} & p(\boldsymbol{x_t}, \boldsymbol{M_t}|\boldsymbol{Z_{t-1}}\boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t}) \\ &\quad = p(\boldsymbol{x_t}|\boldsymbol{M_t}, \boldsymbol{Z_{t-1}}\boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t}) p(\boldsymbol{M_t}|\boldsymbol{Z_{t-1}}\boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t}) \end{aligned}$$

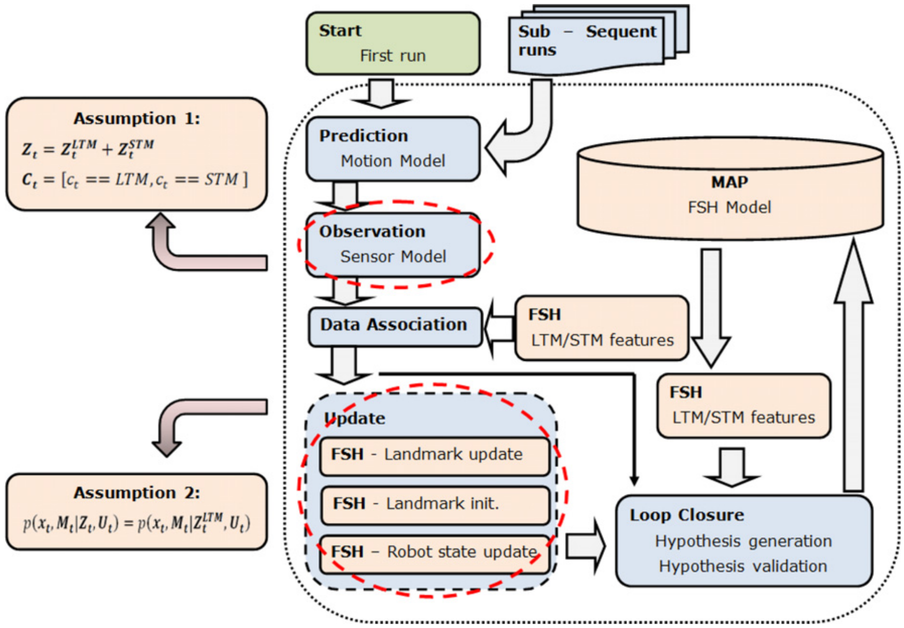

Fig. 2. FSH model integrated in the SLAM framework.

$$= \int p(\boldsymbol{x_t}|\boldsymbol{x_{t-1}}, \boldsymbol{M_t}, \boldsymbol{Z_{t-1}}\boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t})p(\boldsymbol{x_{t-1}}|\boldsymbol{M_t}, \boldsymbol{Z_{t-1}}\boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t})$$
$$\times\, p(\boldsymbol{M_t}|\boldsymbol{Z_{t-1}}\boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t})d\boldsymbol{x_{t-1}}$$
$$= \int p(\boldsymbol{x_t}|\boldsymbol{x_{t-1}}, \boldsymbol{u_t})p(\boldsymbol{x_{t-1}}, \boldsymbol{M_t}|\boldsymbol{Z_{t-1}}\boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_{t-1}})d\boldsymbol{x_{t-1}}. \tag{5}$$

The last line in Eq. (5) depicts the well known SLAM prediction step, considering only LTM correspondence variables. The final filter equation is obtained by expanding Eqs. (4) and (5) into Eq. (3). This is shown in Eq. (6):

$$p(\boldsymbol{x_t}, \boldsymbol{M_t}|\boldsymbol{Z_t}, \boldsymbol{U_t}, \boldsymbol{C_t^{LTM}}) \propto p(c_t = LTM)p(\boldsymbol{z_t}|c_t = LTM, \boldsymbol{x_t}, \boldsymbol{M_t})$$
$$\times \int p(\boldsymbol{x_t}|\boldsymbol{x_{t-1}}, \boldsymbol{u_t})p(\boldsymbol{x_{t-1}}, \boldsymbol{M_t}|\boldsymbol{Z_{t-1}}\boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_{t-1}})d\boldsymbol{x_{t-1}}. \tag{6}$$

Eq. (6) shows the SLAM posterior weighted by the term $p(c_t = LTM)$, which is the likelihood that an observation $\boldsymbol{z_t}$ corresponds to an LTM feature. Exploiting the fact that c_t is conditionally independent of $\boldsymbol{z_t}$ given $\boldsymbol{x_t}$, $p(c_t = LTM)$ is extracted from the normalized FSH and regarded as a probability distribution. FSH values are related to the information content of a landmark, in this work they are computed by the sum of the reciprocals of the main diagonal elements of the covariance matrix, as suggested in [29] and depicted in Eq. (7):

$$fsh(\boldsymbol{z_{t,i}}, \boldsymbol{\Sigma_{z_t}^i}) = \sum_{n=1}^{R} \frac{1}{\sigma_{nn}^2} \tag{7}$$

where $\boldsymbol{z_{t,i}}$ is the i-th feature of the measurement performed at time t, R is the rank of $\boldsymbol{z_{t,i}}$ and σ_{nn}^2 is the n-th value of the covariance matrix $\boldsymbol{\Sigma_{z_t}^i}$. Eq. (7) is part of the rehearsal process depicted in Fig. 1, which is in charge of rating the map landmarks when FSH values are updated. As described in Section 3.1, there are other options to define the feature strength, e.g. the Hessian value in the SURF descriptor [27] or the matching distance. Then, Eq. (7) can be replaced by any of these options.

The aim of the probabilistic derivation done from Eqs. (3) to (7) is to show that the FSH model proposed can be used in different SLAM methods. The right term of Eq. (6) is basically the SLAM formulation, which considers LTM features as observations in the filtering process. This means that parametric (e.g. EKF) and non-parametric filters (e.g. particle filters) are both suitable for the FSH model.

3.2.2. LTM/STM feature classification and STM features removal

3.2.2.1. LTM/STM feature classification. The FSH model discussed up to now considers two types of features, namely LTM and STM. Taking into account the FSH model depicted in Fig. 1 and the human memory model proposed by [22], it is important to define a discrimination method to classify the environmental features as either LTM or STM. As a result, useless STM features can be removed in order to increase the SLAM algorithm scalability. In our previous published work [7], the feature classification method was based on a fixed threshold computed statistically. In this work, however, the map landmarks are classified automatically as LTM or STM each time the SLAM process completes one iteration.

The LTM/STM feature classification process can be summarized as follows: first, after each SLAM iteration measured features and their associations are available; second, using their strength (Eq. (7)) the normalized FSH model is built (see Fig. 3(a)) and sorted in descending order; finally, clustering into LTM or STM sets is performed through *k-means* [28].

The LTM/STM feature classification is a clustering process with low dimensionality (rank-2), for this reason keeping the number of data points low is desirable. In this work, a global stochastic map is used as environmental representation. Then, a new node is started when the number of features in the current node reaches a maximum [6]. Using this strategy a low number of features for each node of the topological map are analyzed. However, other sub-mapping techniques can be used [6].

Formally, given a set of observations $(sf_1, sf_2, \ldots, sf_N)$ where each observation is the feature strength, the aim of clustering is to classify the N features into LTM or STM, $\boldsymbol{S} = \{S_{LTM}, S_{STM}\}$. The within-cluster sum of squares cost function yields as depicted in Eq. (8):

$$\min_{\boldsymbol{S}} \sum_{j=1}^{K} \sum_{i=1}^{N} \|\boldsymbol{sf}_i - \boldsymbol{\mu_j}\|^2 \tag{8}$$

where K is the number of sets which in this case is 2, and μ_j is the mean of the data points belonging to the set S_j (j stands for LTM or STM). Once the minimization of Eq. (8) is done, all those features which belong to the set with highest mean are considered as LTM.

Fig. 3(a) shows a typical normalized FSH where the data points depicted using diamonds correspond to LTM features, and those

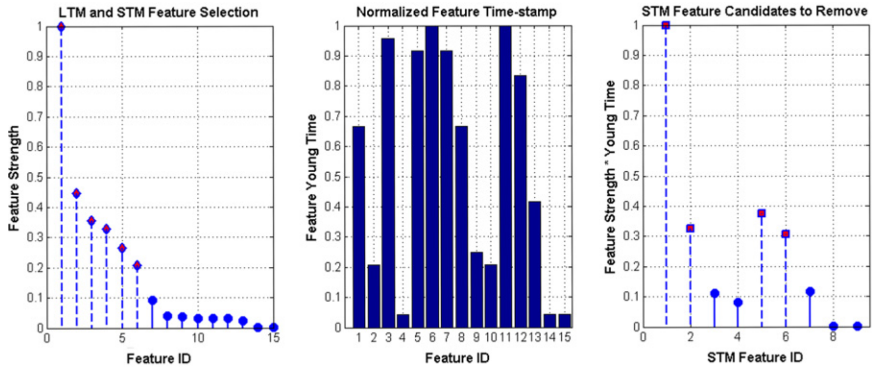

Fig. 3. (a) LTM and STM feature selection using *k*-means. (b) Normalized feature time stamp with respect to the current viewing step. (c) STM feature candidates to be removed (circle-shaped).

points drawn with circles correspond to STM features. This classification was done applying Eq. (8) to FSH values, which is directly related to feature strength. It is worth noting that a minimum number of features are required in order to perform clustering using *k-means* with at least two sets. As a rule of thumb the number of sets can be estimated by Eq. (9), justified in [30]:

$$k \approx \sqrt{\frac{N}{2}} \tag{9}$$

where N is the number of data points. According with Eq. (9), $N = 8$ is the minimum number of features to perform clustering with *k-means*. However, when the mobile robot is visiting a new area may be the number of features observed could be less than eight. In this case, the LTM/STM classification is performed using exponential fitting in order to estimate the mean lifetime as the threshold value to distinguish LTM from STM features. The exponential fitting of the FSH is performed using a direct minimization method based on the Nelder–Mead simplex algorithm described in [31], which tries minimizing the sum of squares error described in Eq. (10).

$$f(A, \lambda) = \sum_{i}^{K} (Ae^{-\lambda x_i} - y_i)^2 \tag{10}$$

where K is the number of data points, A and $\boldsymbol{\lambda}$ are the parameters to estimate and (x_i, y_i) are the data point pairs. Once the exponential decay parameters function are estimated, the LTM/STM feature classification is performed selecting the mean lifetime as the threshold value to differentiate between these two sets. The mean lifetime is shown in Eq. (11).

$$\tau = Ae^{-1} \tag{11}$$

3.2.2.2. STM features removal. Observing Fig. 3(a), particularly STM features, their strengths are comparatively low with respect to LTM feature strength. This is can be due to two main reasons: either the features have been recently incorporated to the map and their strength is not high, or the features are old and they have not been re-observed inducing a relative decreasing of their strength. Taking advantage of this, an STM feature removal algorithm can be proposed by considering the following requirements: limiting the ever-growing number of STM features, preserving the newest features even though they have low strengths, and removing the oldest and weakest features.

The STM feature removal process can be summarized as follows: first, once the LTM/STM classification process finishes the STM features can be identified; second, our approach estimates how old a feature is with respect to the first time the map was created; third, clustering again the STM features in two sets those STM features to be removed and those features to preserve, to do so the joint given by the feature strength and their age is considered; finally, all those features belonging to the STM features to be removed set are deleted from the current map.

The STM feature removal first step was described in Section 3.2.2.1. To estimate how old a feature is, the FSH model considers a time stamp associated to each feature in the environment map. This time stamp shows how many times a feature has been observed, however this value may be different from the number of times the robot has visited the feature's surrounding area. For this reason, the former is normalized with respect to the latter. Fig. 3(b) shows the normalized time stamp corresponding to those features in Fig. 3(a). The higher the histogram values the younger the feature.

Observing the normalized time stamp values for features 7–15, the weakness of features 11–14 is not justification enough to remove these features since they are new. On the other hand, features 14 or 15 could be candidates for removal, since they have low strengths and they are the oldest. Formally, assuming that the normalized feature time stamp and the feature strength stored in the FSH are independent given the robot position, the likelihood model for the STM feature removal is given in Eq. (12):

$$p(sf_i^{STM}, tf_i^{STM} | \boldsymbol{x_t}) = p(sf_i^{STM} | \boldsymbol{x_t}) p(tf_i^{STM} | \boldsymbol{x_t}) \tag{12}$$

where $\boldsymbol{x_t}$ is the robot position, sf_i^{STM} is the i-th STM feature strength and tf_i^{STM} is the i-th STM feature time stamp. Fig. 3(c) shows the resulting likelihood model given by Eq. (12) for the typical FSH model and time stamps of Fig. 3(a) and (b). Here, the classification problem is to group STM features into two sets: those STM features to be removed and those to be preserved. Finally, the *k-means* algorithm is used as depicted in Eq. (8) doing the appropriate changes in the variables. As a result, the circle shaped points in Fig. 3(c) show the STM features to be removed. In the case the number of STM features were not enough to cluster them ($N < 8$), it is a better option gather more evidence about STM features before taking the decision of delete them. Therefore, if $N < 8$ the STM features removal process does not take place.

3.2.3. Localization and map building

Previous sections described the essential characteristics of the FSH model such as: the structure inspired on the human memory model, the probabilistic foundation to adapt the FSH model to the current SLAM solutions, the FSH model rehearsal procedure to promote features from STM to LTM, the STM/LTM feature classification method and the STM feature removal. However, considering Fig. 2 additional insights have to be described in order to integrate the FSH model implementation into any SLAM solution.

The following assumptions have to be considered:

1. the observations can be decomposed into LTM and STM, as depicted in Eq. (1)
2. the posterior of the robot pose depends on the LTM features only, as depicted in Eq. (2).

3. the FSH model is computed once the SLAM process has finished the update stage.
4. the LTM/STM feature classification is performed at each time step; however the STM feature removal is performed each time a previously mapped area is re-visited. The latter includes: loop closure situations or further SLAM runs.

According to Fig. 2, the FSH model is involved in the data association, map and state update and loop closure detection processes. Algorithm 1 describes how the FSH model can be integrated into current SLAM methods. Our implementation of the FSH model is based on the FastSLAM algorithm [8]; in this context, each particle has a separate FSH model, avoiding the map consistency inconvenient when the best particle switches. However, Algorithm 1 considers a general notation introduced in Table 2 with the aim of easily integrate it on current SLAM methods.

In Algorithm 1, the SLAM prediction and measurement stages are not affected by the FSH model (lines 2–6), though, the data association process does keep in mind current LTM and STM feature indexes in order to perform its work hierarchically (line 8), i.e., LTM features are associated before STM features. Data association of STM features is important because depending on the FSH, STM features can increase the strength and eventually become part of LTM.

Another SLAM stage affected is the filter update process (line 10). Here LTM features are considered only to correct the robot pose and covariance. On the other hand, re-observed STM features are updated in order to estimate their covariance matrix and in this way compute their strength. Once the SLAM filter finishes, the FSH model is computed using the data association vector, current FSH values and the updated covariance matrices of the re-observed features (lines 13–18). Subsequently, LTM/STM feature classification take place as well as the STM features removal process (lines 19–23).

LRF sensors are popular in the robotics community. The FSH model described so far can be applied to any type of observation. In this work, the laser scan is also processed by the FSH model (line 24). By doing so, a more stable local environmental structure can be obtained, filtering out pedestrians and moving objects. Formally, given a sequence of LRF readings $\boldsymbol{S_t} = \{\boldsymbol{s_1} \quad \cdots \quad \boldsymbol{s_N}\}$, where N is the total number of LRF scans and $\boldsymbol{s_j}$ corresponds to an $m \times N$ matrix being m the rank of the data points, a set of votes vs_j^i can be computed for each i-th point in the j-th laser scan.

Using the previous filtered robot positions the $\boldsymbol{s_{t-1}}$ LRF readings are sequentially registered, yielding a local map patch $\boldsymbol{PS_{t-1}}$. At each step, the $\boldsymbol{PS_{t-1}}$ is aligned with the current LRF reading $\boldsymbol{s_t}$ using the filtered robot position $\boldsymbol{x_{t+1}}$ and the set of votes for the FSH model is computed using the Nearest-Neighbor (NN) approach. The vote of a data point is defined as depicted in Eq. (13):

$$vs_j^i = \begin{cases} 1 & \text{if } \|\boldsymbol{s_t^i} - \boldsymbol{PS_{t-1}^k}\|^2 < \mathit{LRFresThreshold}^2 \\ 0 & \text{Otherwise} \end{cases} \tag{13}$$

where $\boldsymbol{s_t^i}$ is the i-th point in the current laser scan, $\boldsymbol{PS_{t-1}^k}$ is the nearest-neighbor k-th point in the previous map patch, and *LRFresThreshold* depends on the LRF resolution. In this work, the URG-04LX LRF is used and its range resolution is 0.04 m [32], thus the *LRFresThreshold* value was set to 0.04 m.

Detecting loop closure situations is a challenging task. Many loop-closure detection techniques have been proposed [33]. On one hand, map-to-map matching methods are based on finding correspondences between common features in different sub-maps [34]. Furthermore, image-to-image matching methods detect loop-closures based on recognizing visual appearance of places [21]. Finally, image-to-map or feature-to-map matching methods are based on finding correspondences between features and maps [35,36]. Once the loop-closure is detected, topological representations of the environment are helpful to improve the map and robot position estimation [37]. In this work, a global stochastic map representation is stored using the relative locations between nodes. A new node is started when the number of features in the current node reaches a maximum, or when no correspondences were found by the data association stage.

There are two ways in which the FSH model is used for loop closure detection: first, the re-visited map patch $\boldsymbol{PS_j}$ and the current scan $\boldsymbol{s_t}$ are used to get a similarity measure using the Hausdorff fraction [38], which is a metric used to measure the distance between two sets of points; second, an overlapping of 60% [39,40] in the LTM features is used between the current node and the re-visited node. In this work, an overlapping of 60% (both LRF map patches and landmarks) is used to issue a loop-closure alert (line 27). In this case, a set of conditions such as the robot position, the base position of the node that closes the loop and a shared LTM feature are used to constrain the loop.

Finally, in line 29 the detected loop closure is processed using the graph representation of the robot poses, their uncertainties, and the additional constraint over the observed LTM feature. The non-linear optimization is done using TORO [41], which has been

Algorithm 1. SLAM and FSH model computation.

1. ***while*** operating
2. **% SLAM prediction.**
3. $[\widehat{\boldsymbol{x_t}} \;\; \widehat{\boldsymbol{Px_t}}] = doSLAMprediction(\boldsymbol{x_{t-1}}, \boldsymbol{u_t}, \boldsymbol{Px_{t-1}}, \boldsymbol{R_t})$;
4. **% SLAM measurement.**
5. $\boldsymbol{Z_t} = doMeasurement(\widehat{\boldsymbol{x_t}}, \widehat{\boldsymbol{Px_t}}, \boldsymbol{M_t}, \boldsymbol{Q_t})$;
6. $\boldsymbol{s_t} = getCurrentScan()$;
7. **% SLAM data association.**
8. $\boldsymbol{H_t} = getDataAssociationVector(\boldsymbol{Z_t}, \boldsymbol{M_t}, \boldsymbol{Zind_t^{STM}}, \boldsymbol{Zind_t^{LTM}})$;
9. **% SLAM update.**
10. $[\boldsymbol{x_{t+1}}, \boldsymbol{Px_{t+1}}, \boldsymbol{Z_{t+1}}, \boldsymbol{M_{t+1}}] = doSLAMupdate(\widehat{\boldsymbol{x_t}}, \widehat{\mathbf{Px}}_{\boldsymbol{t}}, \boldsymbol{M_t}, \boldsymbol{Z_t}, \boldsymbol{H_t}, \boldsymbol{FSH_t}, \boldsymbol{Zind_t^{STM}}, \boldsymbol{Zind_t^{LTM}})$;
11.
12. **% Current FSH model.**
13. $\boldsymbol{FSH_t} = getFSHvalues()$;
14. **% FSH model computation—Rehearsal process.**
15. **for** $\boldsymbol{z_{G,t+1}^i} = \boldsymbol{Z_{t+1}}(\boldsymbol{H_t})$
16. $\quad \boldsymbol{Sz_{G,t+1}^i} = getFeatureStrength(\boldsymbol{Pz_{G,t+1}^i})$;
17. $\quad \boldsymbol{FSH_{t+1}} = updateFSHvalues(\boldsymbol{Sz_{G,t+1}^i}, \boldsymbol{FSH_t})$;
18. ***end***
19. **% FSH model computation—LTM/STM feature classification.**
20. $[\boldsymbol{Zind_{t+1}^{LTM}}, \boldsymbol{Zind_{t+1}^{STM}}] = doLTM\_STM\_Classification(\boldsymbol{FSH_{t+1}})$;
21. **% FSH model computation—STM pruning**
22. $doSTMfeaturePruning(\boldsymbol{FSH_{t+1}}, \boldsymbol{Zind_{t+1}^{STM}})$;
23. **% FSH model computation—LRF readings.**
24. $\boldsymbol{PS_t} = doFSHoverLRFreadings(\boldsymbol{PS_{t-1}}, \boldsymbol{s_t})$;
25.
26. **% Loop-closure detection.**
27. LC_Alert $= doLoopClosureDetection(\boldsymbol{x_{t+1}}, \boldsymbol{Px_{t+1}}, \boldsymbol{M_{t+1}}, \boldsymbol{H_t}, \boldsymbol{Zind_{t+1}^{STM}}, \boldsymbol{Zind_{t+1}^{LTM}}, \boldsymbol{PS_t})$;
28. ***if*** (LC_Alert)
29. $\quad doConstraintLoop(\boldsymbol{x_{t+1}}, \boldsymbol{Px_{t+1}}, \boldsymbol{M_{t+1}}, \boldsymbol{H_t}, \boldsymbol{Zind_{t+1}^{LTM}})$;
30. ***end***
31. ***end***

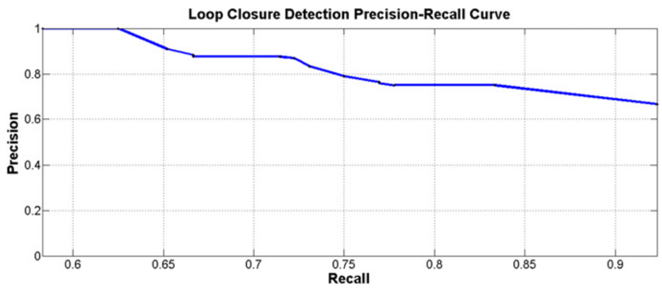

Fig. 4. Precision–recall curve for loop closure detection using the FSH model. The total robot path used to compute this curve was 1.24 km approximately.

adapted to work in incremental mode and through Matlab MEX files.

An important metric to test our loop closure detection approach is through the precision–recall curve, where precision is defined as the number of correct loop closure matches divided by the total number of matches, and recall as the number of correct loop closure matches divided by the total number of expected matches. These definitions are observed in Eq. (14).

$$Precision = \frac{TP}{TP + FP}, \qquad Recall = \frac{TP}{TP + FN} \tag{14}$$

where TP, FP and FN are true positives, false positives and false negatives respectively. Expected matches are defined as previously selected locations in the map within 2 m distance to the current robot location. It is worth remembering that our LRF has maximum range of 4 m. Fig. 4 shows the precision–recall curve for loop closure detection using the FSH model. To compute this curve, fifteen runs over the ground floor were used. The ground floor map contains essentially three loop closure situations, and it is worth noting that the SLAM runs used belongs to our dataset which considers changes in illumination, season in the year, and occlusions due pedestrians. Observing Fig. 4, it reports over 70% precision over more than 80% of the recall range.

4. Sensor model

The type of perception system and the feature extraction method used determine how the environment is represented, how likely it is to re-observe the environment features and how the uncertainty is handled since it depends on the robot sensors used (or a combination of them). One of the most important problems in SLAM is to find correspondences between the observations taken at different places in the environment. These correspondences are crucial to simultaneously estimate the robot position and the map of the environment. For this reason, the environmental feature representativeness and the matching process reliability are important factors of SLAM.

In this work, a sensor model based on the extrinsic calibration between an LRF and an omnidirectional camera [9] is used in order to extract the 3D position of vertical edges in indoor environments [42]. Table 3 summarizes the omnidirectional camera intrinsic parameters and the extrinsic calibration parameters between the Hokuyo LRF and the omnidirectional camera used. Vertical edge features are predominant in indoor structured environments, and they are not deformed by the non-linear distortions introduced by the omnidirectional camera mirror. For self-containment, Algorithm 2 shows the basic steps to extract vertical edge features. It consists of six stages: first, the LRF segments and the vertical edges [43] are detected (lines 1–3); second, using the extrinsic calibration between the LRF and the omnidirectional camera [9], and the Barreto conic projection model [44], the LRF segments are projected onto the sphere and the image plane (lines 6–11); third, the LRF corner uncertainties are found on the image plane (lines 14–16); fourth, the vertical edge model on the image plane is computed and their intersections found with the conics corresponding to the LRF segments (lines 19–24); fifth, using the corner uncertainties in the image plane and the intersects computed above, the Joint Compatibility Branch and Bound (JCBB) [45] test is used to robustly associate each corner with the vertical edge intersect (line 25); finally, the vertical edge range-bearing measurement model is found using the associated LRF corner with respect to the camera frame (lines 26–30).

Table 3
Omnidirectional camera intrinsic calibration parameters, and extrinsic calibration with the Hokuyo LRF.

Variable	Description
$\boldsymbol{x}_{t-1}, \hat{\boldsymbol{x}}_t, \boldsymbol{x}_{t+1}$	Previous, estimated and corrected robot pose.
$\boldsymbol{Px}_{t-1}, \hat{\boldsymbol{Px}}_t, \boldsymbol{Px}_{t+1}$	Previous, estimated and corrected robot pose covariance matrix.
$\boldsymbol{u}_t$	Current motion command.
$\boldsymbol{R}_t, \boldsymbol{Q}_t$	Process and observation uncertainty matrix.
$\boldsymbol{M}_t, \boldsymbol{M}_{t+1}$	Current and updated map.
$\boldsymbol{FSH}_t, \boldsymbol{FSH}_{t+1}$	Current and updated FSH values.
$\boldsymbol{Zind}_t^{LTM}, \boldsymbol{Zind}_t^{STM}$	Current LTM and STM feature indexes.
$\boldsymbol{Zind}_{t+1}^{LTM}, \boldsymbol{Zind}_{t+1}^{STM}$	Updated LTM and STM feature indexes.
$\boldsymbol{H}_t$	Current data association vector.
$\boldsymbol{Z}_t, \boldsymbol{Z}_{t+1}$	Estimated and corrected re-observed set of features.
$\boldsymbol{Z}_t^{LTM}, \boldsymbol{Z}_t^{STM}$	Current LTM and STM landmarks.
$\boldsymbol{z}_{G,t+1}^{i}$	Corrected i-th feature, G stands for LTM or STM.
$\boldsymbol{Pz}_{G,t+1}^{i}$	Corrected i-th feature covariance matrix, G stands for LTM or STM.
$\boldsymbol{Sz}_{G,t+1}^{i}$	i-th feature strength, G stands for LTM or STM.
$\boldsymbol{PS}_{t-1}, \boldsymbol{PS}_t$	Previous and computed FSH model for the laser scans.
$\boldsymbol{Z}_{0:t}^{LTM}, \boldsymbol{PS}_{0:t}^{LTM}, \boldsymbol{A}_{0:t}^{LTM}$	LTM landmarks, FSH model for the laser scans and appearance-based image descriptors from $t = 0$ to $t = t$.

Fig. 5(a) shows an example of the resulting data association process on the image plane described above. In this figure, the conic intersects are shown in circle-shaped points and the associated LRF corners are shown in star-shaped points. The remaining LRF corners are also shown in diamond-shaped points. Fig. 5(b) shows the consistency between the laser scan matching and the corresponding vertical edges in the scene using the sensor model described above. The vertical edges depicted in this figure correspond to the most predominant ones. The vertical edge position uncertainties are also shown, as well as their measured lengths.

5. Experiments and results

The experimental evaluation of this work was carried out at the University of Girona using an indoor dataset captured over one year at different times of day and seasons of the year. This was done due to the lack of publically available datasets of dynamic environments. The more recent work in this sense is the COLD database [46], but the detailed intrinsic and extrinsic calibration parameters of the sensors involved are not available, which did not allow the sensor model presented in Section 4 to be tested.

Our dataset shares three important properties with the COLD database: first, most of changes are due weather conditions, it means robot runs were performed in different seasons of the year (winter, spring, summer and autumn); second, changes due times of day were also considered, it means robot runs were performed at morning, afternoon and night; third, pedestrians were not warmed about the robot runs, even though robot runs were performed when many people were walking around. Sadly, big structural changes are not common, however moving doors are more often and they involve visual and structural changes in the environment. We believe that these changes are important to test long-term SLAM methods.

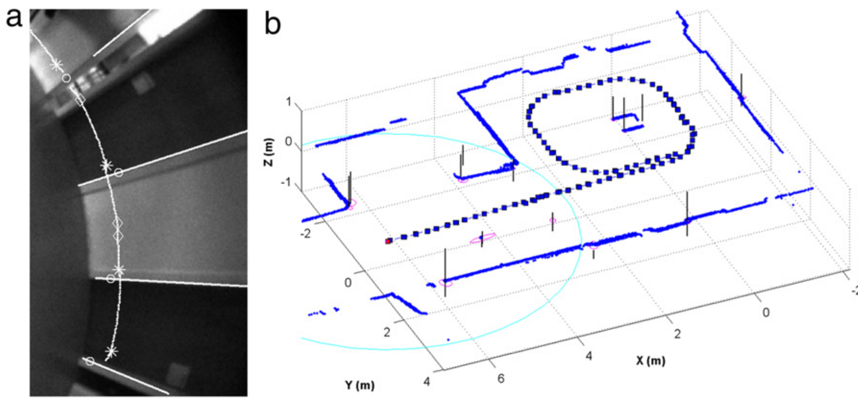

Fig. 5. (a) Vertical edges and their LRF corner associations on the image plane. (b) Detailed view of the predominant vertical edges in the map.

Algorithm 2. Vertical edge feature extraction.

```
lrfSegments = getLRFsegments();
lrfCorners = getCornersFromSegments();
veListFOV = getCatadioptricVerticalEdgesInFOV();

% Computing the segments normal in the sphere and
  their conic matrix on the image plane.
lrfSegNormals ← []; lrfSegConic ← [];
for each S ∈ lrfSegments
    lrfSegSph = doProjectSegment2Sphere(S);
    lrfSegNormals ← getNormalVector(lrfSegSph);
    lrfSegConic ← getBarretoConicProjection(lrfSegSph);
end


% Computing the corners uncertainty on the image
  plane.
[Jp, Jr, Js, Jd, Ji] = doPropagateCornersUncertainties
                        (lrfCorners);
sensorSigma = getSensorSigma(lrfCorners);
lrfCornersSigma = getUncertaintyOnImagePlane
                        (Jp, Jr, Js, Jd, Ji, sensorSigma);

% Vertical edge intersection with the LRF segment
  conics on the image plane.
veMeasurement ← [];
for each ve ∈ veListFOV
    veImgModel = doComputingVEmodel(ve);
    for each Ns ∈ lrfSegNormals; Cs∈lrfSegConic
        [conicCenter, conicRadii]
                = getConicCenterRadii(Cs);
        conicInt = getConicIntersects
                (coniCenter, conicRadii, veImgModel);
        H = getCornerIntersectsDataAssociation
            (conicInt, lrfCornersImg, lrfCornersSigma);
        If empty(H)
            continue;
        else
            veMeasurement ← doMeasurement
                        (H, lrfCorners);
        end
    end
end
```

The collected dataset includes seven (7) runs of each floor of Building PIV at the University of Girona, covering a total distance of 550 m, 445 m and 640 m of the ground, first and second floor respectively. Fig. 6 shows the estimated map over the CAD map, and the stochastic topological map of each floor of Building PIV at the University of Girona. As can be observed from Fig. 6, the map's large size can cause visualization problems if a detailed view is needed. For this reason, the topological representation of the environment is included in order to ease the visualization of the results presented in this section. The experimental validation was conducted as follows: first, at each floor the initial map was built using the FastSLAM algorithm [8]; second, seven further SLAM runs were performed for each map. At each run a new map is generated, which considers the respective changes in the environment. This new map is loaded in the next run.

The season of the year and time of day of SLAM runs performed at each floor are described as follows: at ground floor, test were performed using Winter-Afternoon, Autumn-Night, Spring-Afternoon, Autumn-Morning, Winter-Night, Summer-Morning and Summer-Night dataset; at first floor, test were performed using Autumn-Afternoon, Autumn-Night, Spring-Night, Winter-Morning, Winter-Night, Summer-Afternoon and Summer-Morning datasets; at second floor, test were performed using Autumn-Morning, Autumn-Afternoon, Winter-Afternoon, Spring-Morning, Spring-Night, Winter-Morning and Summer-Night datasets.

We tested our approach on a Pioneer 3DX mobile robot equipped with an onboard computer at 1.5 GHz, an omnidirectional vision setup composed of a RemoteReality parabolic mirror with a diameter of 74 mm, a UI-2230SE-C camera with a resolution of 1024 × 768 pixels, and a URG-04LX LRF (Fig. 7(a)). The environmental conditions in which the dataset was collected can be observed in Fig. 7(b), where each row corresponds to each floor in Building PIV. From these omnidirectional images, it can be observed that there are illumination changes and occlusions caused by pedestrians, in this way ensuring real-world experimental conditions.

The experiments conducted to test the FSH model were divided in three parts: first, a static LRF in order to show how the FSH model works with range data, and how the appearance representation of the environment is updated in the presence of dynamic objects (Section 5.1); second, a set of qualitative results, such as filtering dynamic objects, map quality, and map update over the localization and mapping runs (Section 5.2); and third, in the absence of ground-truth, quantitative results involve the measurement of the SLAM performance using the average likelihood of the range scan reading given the estimated robot position [18], and the LTM scan model (Section 5.3). The mean pose error over the robot path subsequently took into account all the mapping and localization runs, which showed a bounded pose error despite the long-term runs,

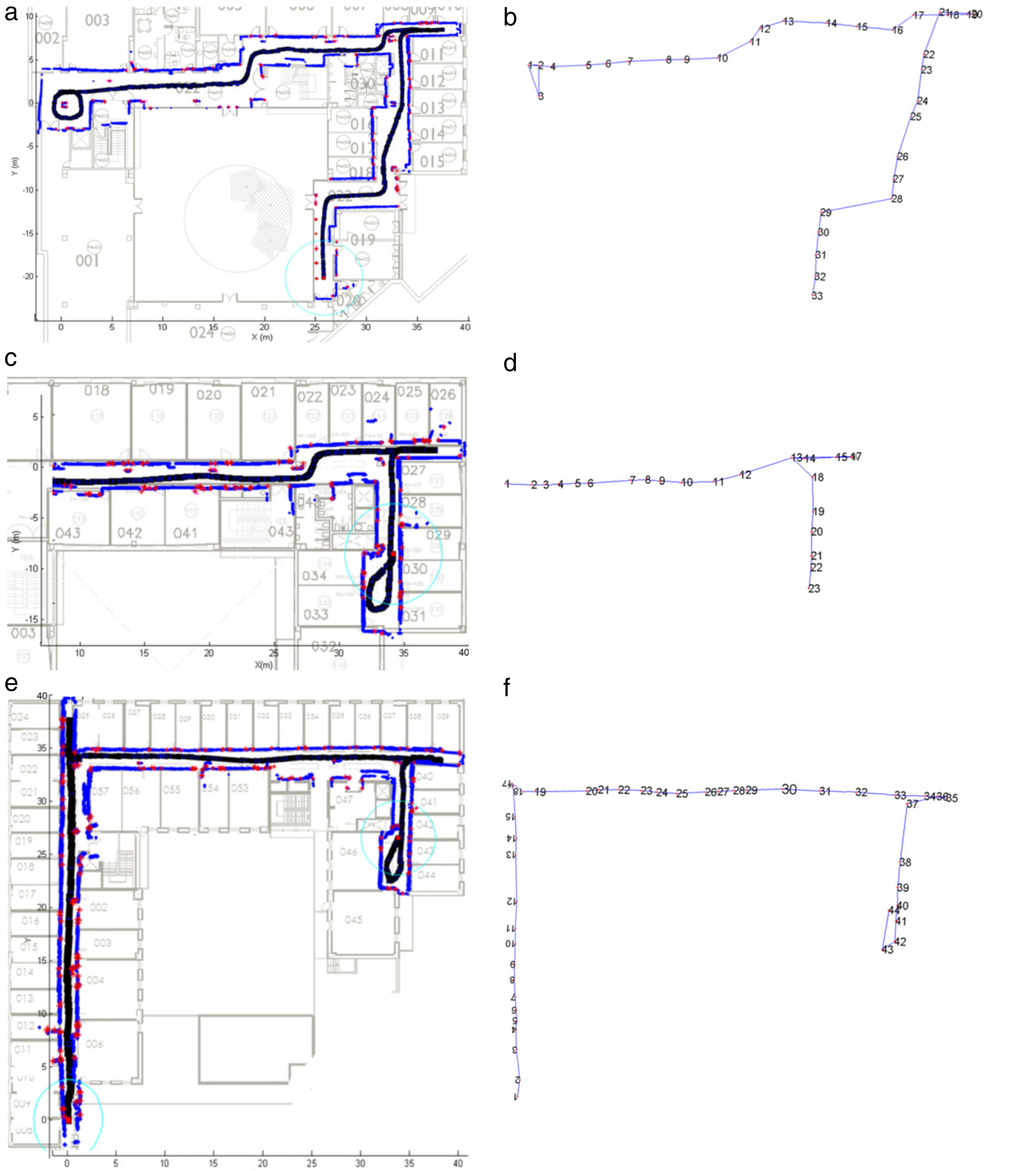

Fig. 6. Building PIV of the University of Girona. (a)–(b) Map, and graph of the ground floor. (c)–(d) Map, and graph of the first floor. (e)–(f) Map, and graph of the second floor.

and the dynamism of the environment. Furthermore, to demonstrate the system scalability, the mean number of LTM and STM features by node over the localization and mapping runs was considered. The aim of this test is to show the performance of STM pruning discussed in Section 3.2.2, and demonstrate that the FSH model can deal with large environments and long periods of operation. Also, we measured the matching effort [14] over the localization and mapping runs; in this work, the matching effort is the mean percentage of LTM or STM matched features. This value is compared to the full matching effort which is measured when the FSH model is not considered.

5.1. Static Laser-based experiment

The static LRF-based experiment took place in a crowded corridor in Building PIV of the University of Girona. Basically, our goal is to test how the FSH model behaves in crowded environments, extracting the most stable environment configuration despite of

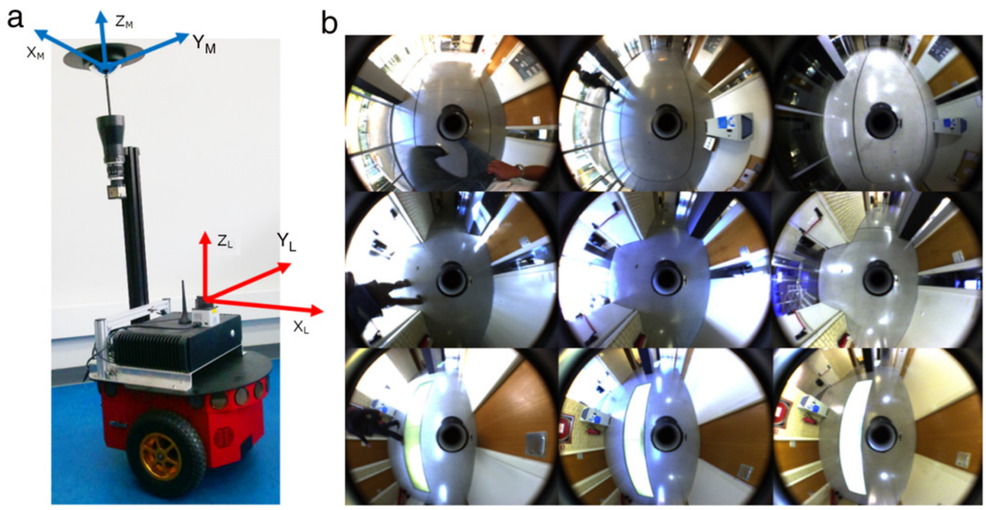

Fig. 7. (a) Pioneer 3DX robot and coordinate frames of the LRF and the omnidirectional camera. (b) Typical omnidirectional images taken from the collected dataset. Each row corresponds to each floor in Building PIV.

pedestrian occlusion. The Pioneer 3DX robot acquired data over 30 min at lunch time. The robot surroundings were composed of static objects (walls, doors and windows). During the data acquisition the appearance of the environment was artificially changed and many pedestrians were passing by. Fig. 8 shows the evolution over time using the FSH model (Fig. 8(a)) and the LTM laser readings (Fig. 8(b)), the lighter the color the less stable the laser reading.

Using the FSH model and Eq. (13), laser readings were continuously classified as LTM or STM. Figs. 8(a) and 7(b) show the time at which box No. 1 was placed modifying the appearance of the environment. The FSH model starts assigning votes to those laser readings belonging to box No. 1, though they are not immediately classified as LTM until the laser reading votes are high enough. When comparing Fig. 8(a) and (b), despite the fact that the corridor selected is very crowded those spurious laser readings corresponding to the dynamic obstacles do not appear in the LTM laser model. In the end, only those environmental changes considered stable by the FSH model were shown in the LTM laser readings.

5.2. Qualitative results

In this section, three qualitative results are presented in order to visually observe the behavior of the FSH model in different situations presented in our dataset such as: filtering dynamic objects (pedestrians), which can cause erroneous robot position estimations and spurious features (Section 5.2.1); the map quality over the mapping and localization runs (each run considers changes in illumination due different times of day and seasons), which gives evidence about the FSH model stability over time (Section 5.2.2); and the capability to update the learnt map when changes in the environment have taken place, which are due to pedestrians and changes in illumination causing erroneous position estimations (Section 5.2.3).

5.2.1. Filtering dynamic objects

Dynamic objects cause basically two main problems:if they are not handled properly, they introduce localization errors [11]; also, they cause spurious features in the map of the environment carrying no information about the vehicle pose estimation [10]. Fig. 9 shows LRF readings and features of the map of the ground level, particularly nodes 1 and 33 of the topological map which are a busy area of Building PIV. The left column of Fig. 9(a) and (b) shows the LRF readings and features without considering the FSH model, and on the right side of Fig. 9 the LTM model of the LRF readings and features is shown. In addition, Fig. 9(a) and (b) also shows the vertical edge uncertainty ellipses as either STM or LTM features, and Fig. 9(c) shows the environmental conditions.

Comparing the left side and right side of Fig. 9, not only are the pedestrians filtered out as a result of applying the FSH model, but the more stable laser readings persist showing a more accurate representation of the environment. Also, it can be observed that a spurious vertical edge was created when the pedestrians were passing by (left side of Fig. 9(a)). Nonetheless, the LTM version of the map does not show this spurious feature (right side of Fig. 9(a)). This situation is repeated in Fig. 9(b), where a pedestrian was found on the robot right side (Fig. 9(c)). In this case, an STM vertical edge feature was created, but it does not appear in the LTM model of the environment because it has not enough strength to be part of the LTM model of the environment. In addition, it is worth noting that these spurious features could have caused registration errors when the laser readings were processed, though the STM features are not considered in the robot pose estimation, and thus do not affect this process.

5.2.2. Map Quality over mapping and localization runs

SLAM solutions are not error-free. This causes erroneous feature position estimations and consequently, erroneous robot position estimations, which in turn cause LRF scan alignment errors. In long-term mapping and localization runs, this situation could probably make the filter diverge [15]. Therefore, an important result is that the FSH map model of the environment is stable over time. Fig. 10 shows only 3 (10(a), (b), and (c)) of the 7 mapping and localization runs performed at node 31 of the second floor. The STM and LTM map model are shown on the left and right side of Fig. 10(a)–(c), respectively. In addition, the omnidirectional images shown in Fig. 10(d) describe the changing environmental conditions, which basically are two such as: changes in illumination and occlusion by pedestrians.

Fig. 10(a)–(c) shows the clear evidence of the difference between STM and LTM features. This can be observed by comparing the uncertainty ellipses drawn on the 2D plane of the laser scan. At the end of the mapping and localization runs, the more stable vertical edges are shown, and they are consistent with the appearance of the environment. Fig. 10(d) depicts the change in the illumination conditions and the moving obstacles. These environmental conditions cause most STM features shown on the left side of Fig. 9(a)–(c). Nevertheless, these features are not present in the

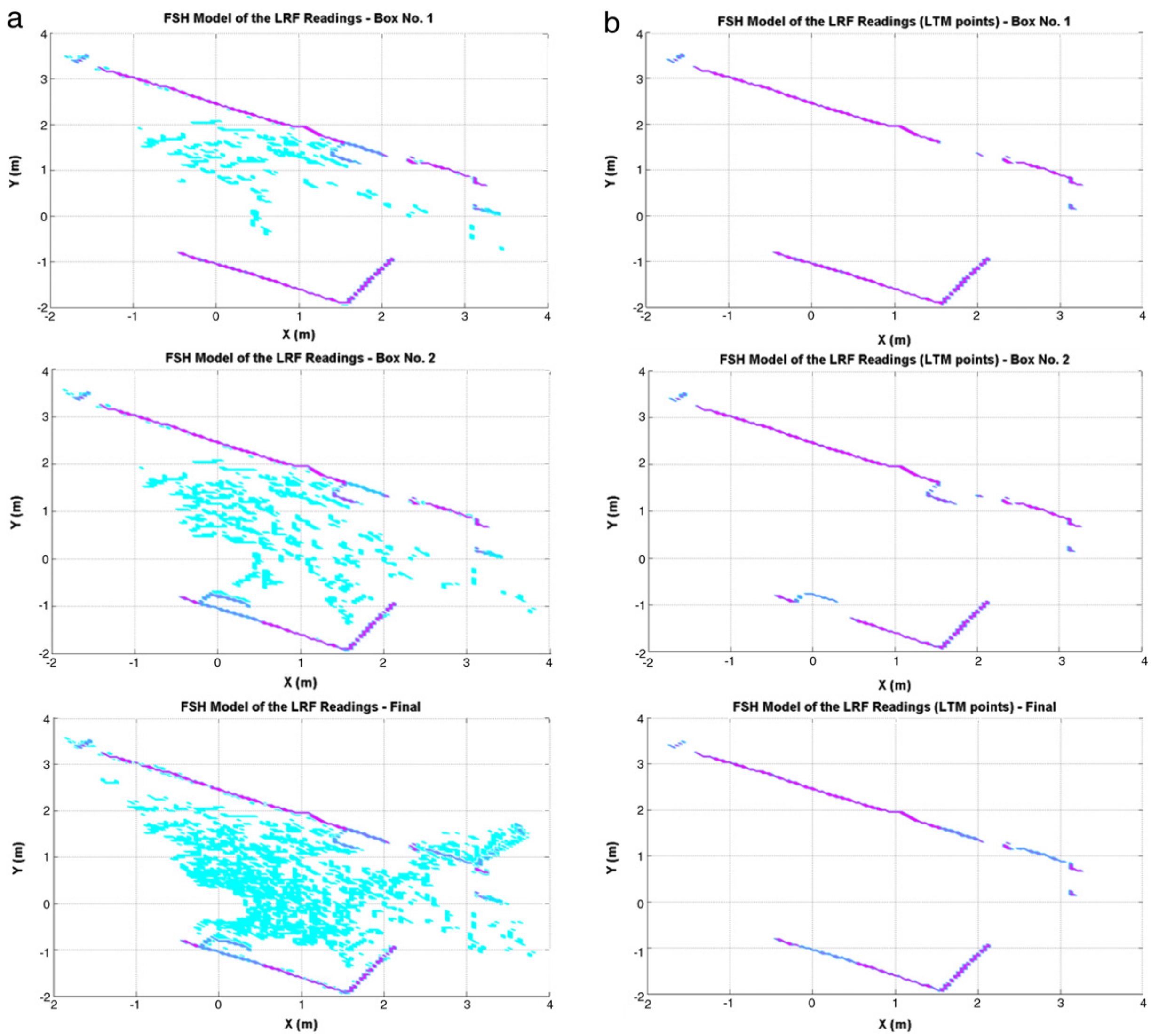

Fig. 8. Static LRF-based experiment. (a) FSH model evolution over time. (b) LTM laser readings evolution over time.

LTM map of the environment. It is worth noting that the straightness of the walls remains consistent despite performing various mapping and localization runs, since only LTM features and LTM laser scans were considered in the robot pose estimation and map representation respectively.

5.2.3. Map update over the mapping and localization runs

Another important result is the capability to update the learned map [16]. In common indoor environments, doors are opened and closed, the furniture is moved and structural changes in the environment are made. In these situations, erroneous robot position estimations occur and it is desirable to update the map accordingly. The FSH model proposed in this work embeds the map update capability into the LTM map version. As a result, the current stable configuration of the environment is learnt over time.

Fig. 11 shows the LTM map at node 5 of the ground floor over 4 mapping and localization runs (Fig. 11(a) of run 1, Fig. 11(b) of run 3, Fig. 11(c) of run 5 and Fig. 11(d) of run 7). Here, it can be observed how the LTM map is continuously modified in order to take into account the state of the door (open/closed). At the beginning this door was closed, but further mapping and localization runs show that the new state of the door is properly updated. In [18] various time constants are considered to hold different environment configurations, then the current observations are matched with these different versions of the environment, and the map version selected is the one which better explains the current measurements. In this work, the FSH model holds one model of the environment namely the LTM map, which embeds the more stable appearance of the environment. This can be observed in Fig. 11(b), where the LTM map shows the state of the door (open/closed). However, this happens temporarily while the LTM map is updated properly, since it is worth recalling that LRF observations are classified between LTM/STM without considering a weight based voting schema, this means that only for LRF readings, inconsistencies as depicted in Fig. 11(b) temporarily happen. However, it can be solved using horizontal lines estimated from raw LRF readings because in this case uncertainty could be measured and used for LTM/STM classification.

Observing Fig. 11(b), in the worst case an object might block half of a hallway and stay there over a considerable period of time, the this object will part of the LTM map. Afterwards, next time the robot re-visits this hallway the object is placed blocking the other half. In this situation, there is a map discrepancy since the corridor appears blocked, therefore the robot could not plan any trajectory along the hallway. Last situation could be solved considering horizontal lines extracted from raw LRF readings to improve the environment model as mentioned above. However, the scope of our work does not consider these extreme situations, and for this reason we mention in conclusions section particularly in future works

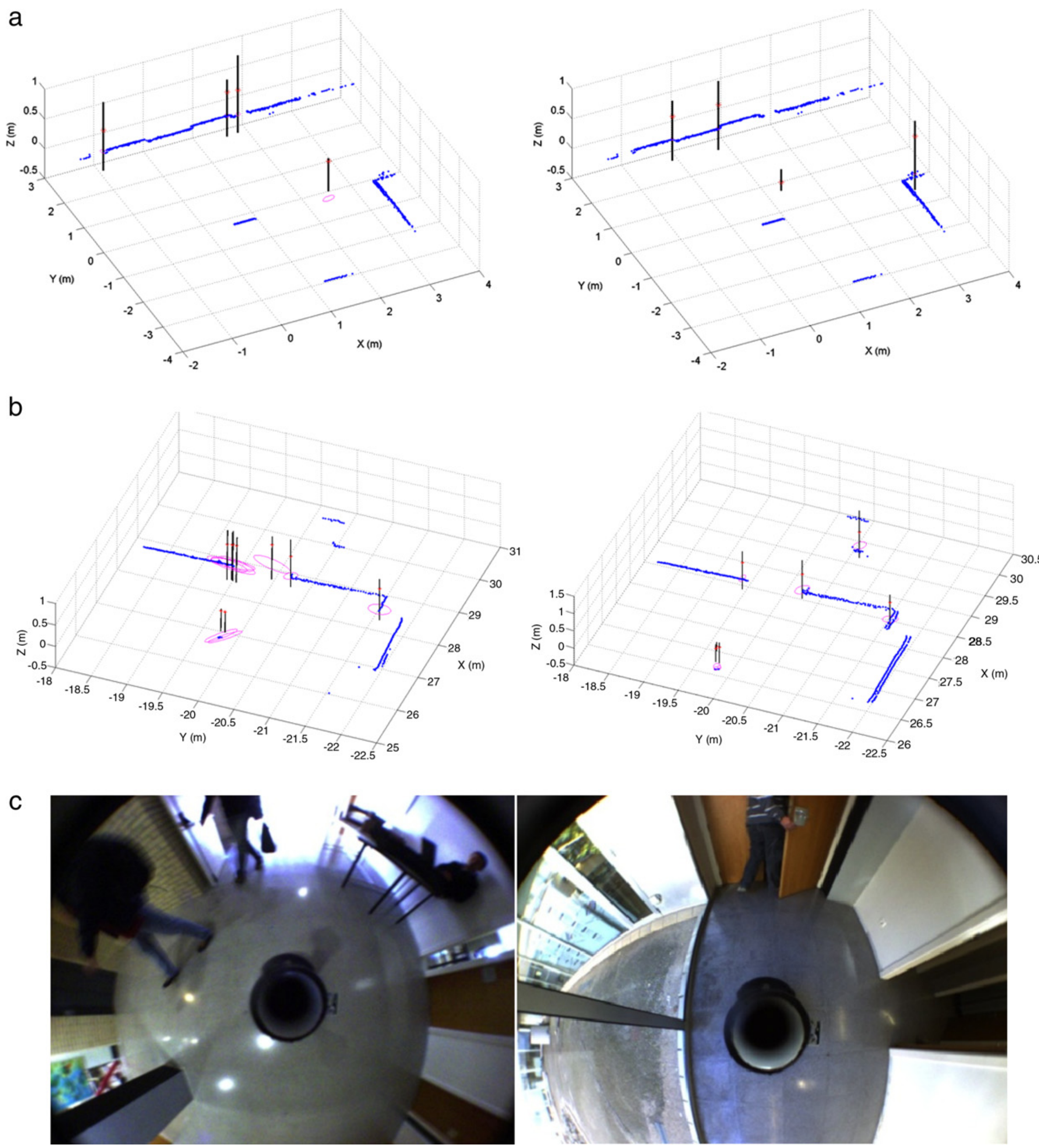

Fig. 9. Filtering dynamic objects. STM (left column) and LTM (right column) features of node 1 (a) and 33 (b) of the ground floor. The environmental conditions are shown in (c).

that better landmark rating methods are needed to resolve temporal map inconsistencies.

Fig. 12 shows the LTM map at node 27 of the ground floor over 4 mapping and localization runs. In this figure a case of map repair is presented. Node 27 on the ground floor is a busy entrance of Building PIV, and in addition the large windows provide considerable illumination changes (see Fig. 12(e)). As a result, the probabilities of making an error in the robot pose estimation increases. This can be observed in the second map update (Fig. 12(b)), where the wall appears at different positions with respect to the first map update (Fig. 12(a)). Further mapping and localization runs rehearse the last hypothesis on the LTM map. For this reason the last mapping and localization run shows the wall in the new position and removes the previous hypothesis. Fig. 12(e) shows four omnidirectional images, each one corresponding to each mapping and localization run. As can be observed, the big window on the right and the pedestrians passing by cause significant changes in the appearance of the environment. These changes have a negative effect in the ground floor map, though when using the FSH model is used throughout the mapping and localization runs, the current stable hypothesis remains.

5.3. Quantitative results

In this section, quantitative results are presented. Measuring the real performance of SLAM solutions requires ground-truth data, which in most cases are hard to obtain. It is even harder to obtain in dynamic environments. In this work, a set of performance measures were selected and the results obtained over all the mapping and localization runs. They considered changes in illumination due different times of day and seasons, occlusion by

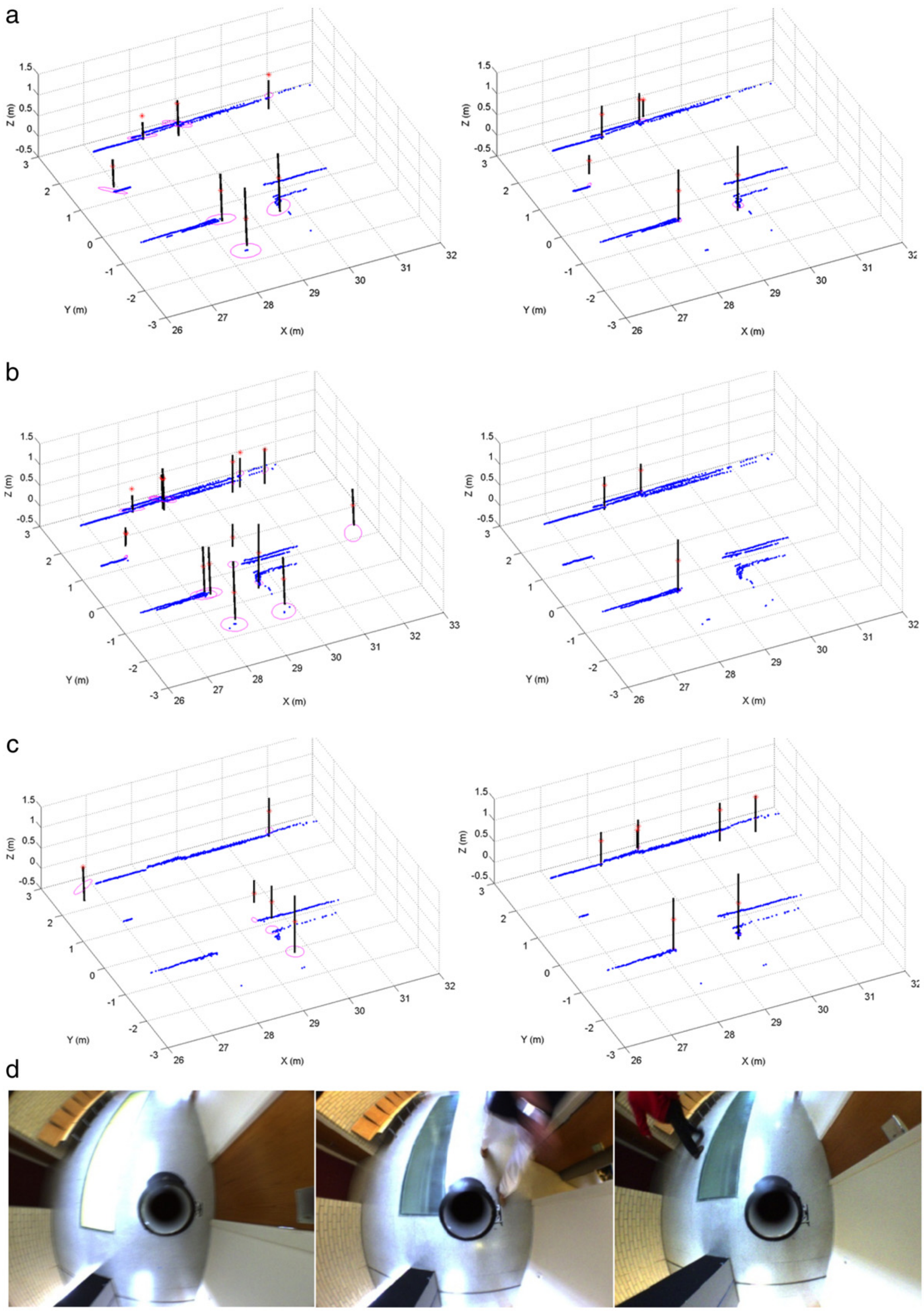

Fig. 10. Map quality over the localization and mapping runs of node 31 on the second floor. (a) Localization and mapping run 1. (b) Localization and mapping run 4. (c) Localization and mapping run 7. (d) Omnidirectional images corresponding to (a)–(c).

pedestrians and structural changes in the environment. Those performance measures are as follows:

1. the average laser scan likelihood over the mapping and localization runs given the robot position estimations and current

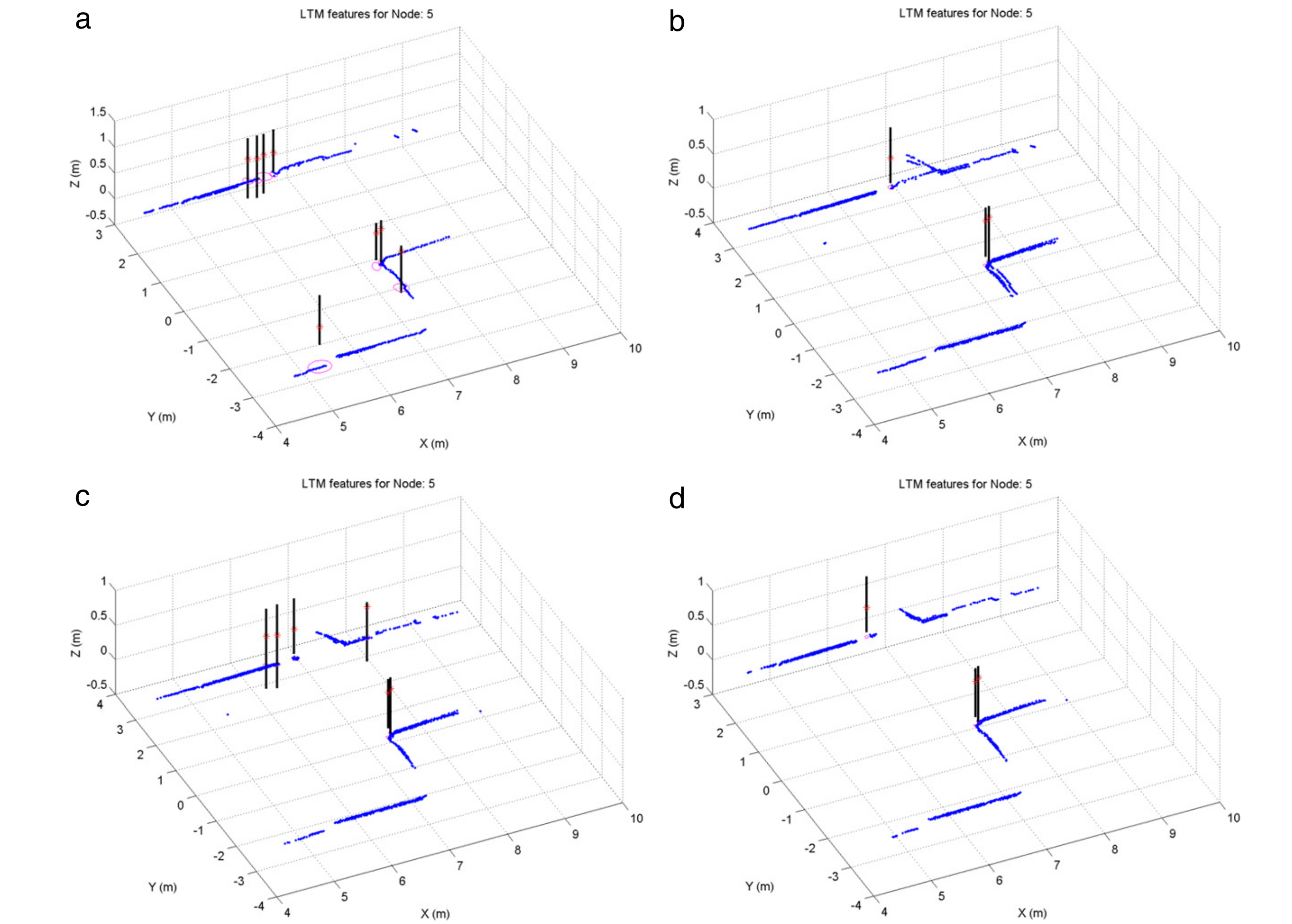

Fig. 11. Map update over the mapping and localization runs. LTM map at node 5 of the ground floor over 4 mapping and localization runs. From top to bottom and left to right: mapping and localization runs 1, 3, 5 and 7.

LTM map. This likelihood is computed using the Hausdorff fraction [38], which is a metric used to measure the distance between two sets of points. Our goal is to measure of how expected the current laser scan is, given the LTM map. It is worth noting that the current laser scan is registered at the current robot position.

2. the mean pose error along the robot trajectory and over the mapping and localization runs. Here, we used the G^2O framework [47] in order to obtain a batch optimization of each mapping and localization run as the ground-truth map.
3. the mean number of LTM, STM and deleted features by node over the mapping and localization runs can provide evidence for the scalability of this work.
4. the mean matching effort over the mapping and localization runs when the FSH model is used (LTM and STM features), and when it is not.

5.3.1. Scan-likelihood over the mapping and localization runs

Fig. 13 shows the mean laser scan likelihood for the ground floor (12(a)), the first floor (12(b)) and the second floor maps (12(c)) over the mapping and localization runs. In all cases, it can be observed that the more mapping and localization runs performed, the higher the scan likelihood when the FSH model is used. The scan likelihood depicted in Fig. 13 is a measure of the localization accuracy given the computed LTM features. The initial scan likelihood in all cases is quite good: 71.31% and 69.73% for the map on the ground and second floor respectively. However, the scan likelihood increases as more mapping and localization runs are performed, meaning that the localization accuracy also increases when the FSH model is used.

The reason for this is because the Hausdorff fraction that directly depends on the number of closest points between two point clouds. To perform this calculation the Hausdorff fraction is based on the nearest neighbor criteria. However, the Hausdorff fraction depends inversely on the number of points in the reference cloud of points. In our case, this is the LTM model. Therefore, it is expected that this measure increases over successive mapping runs because, in first place, the LTM model of the environment contains more information to match with the current laser scan; and in second place, since the LTM model is used for SLAM the average likelihood of laser scans is a measure of how the localization accuracy is improved over time.

The impact of dynamic changes in the environment is shown in Fig. 13. In this figure, it can be observed an increment of the measure uncertainty in the mapping and localization run number 5 at the ground floor, in the mapping and localization runs 3–7 of the first floor and in the mapping and localization run 5 of the second floor. Then, dynamic changes in the environment cause a reduction in the average likelihood of the laser scans.

The highest variation in the scan likelihood uncertainty is present in Fig. 13(b), which corresponds to the first floor map. Here, the SLAM algorithm, in conjunction with the FSH model, faces a great challenge. Observing Fig. 6(d), node 6 on the first floor map has not enough features, and it covers a space of about 6 m,

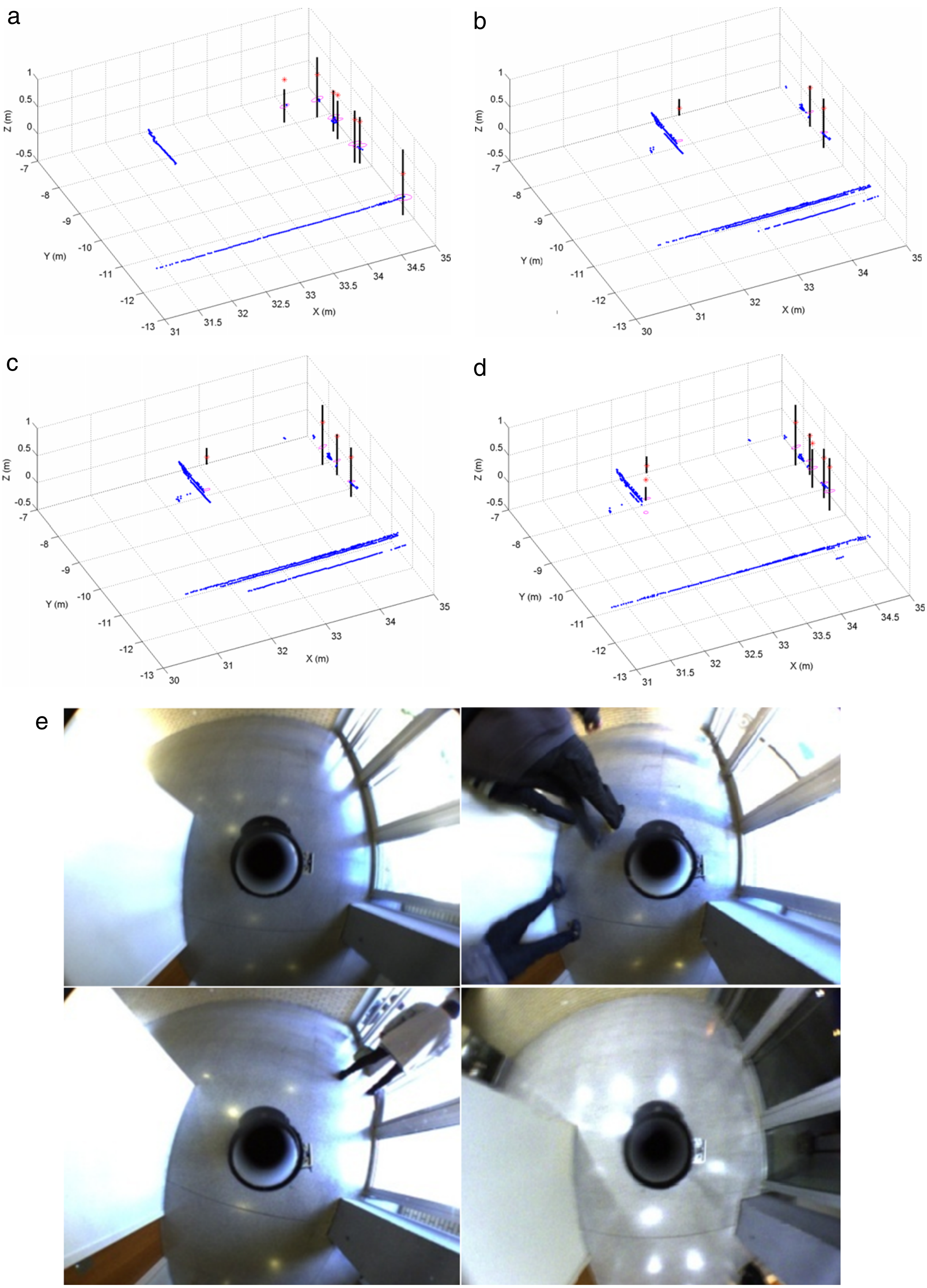

Fig. 12. Map update over the mapping and localization runs. LTM map at node 27 of the ground floor over 4 mapping and localization runs. From top to bottom and left to right: mapping and localization runs 1, 3, 5 and 7.

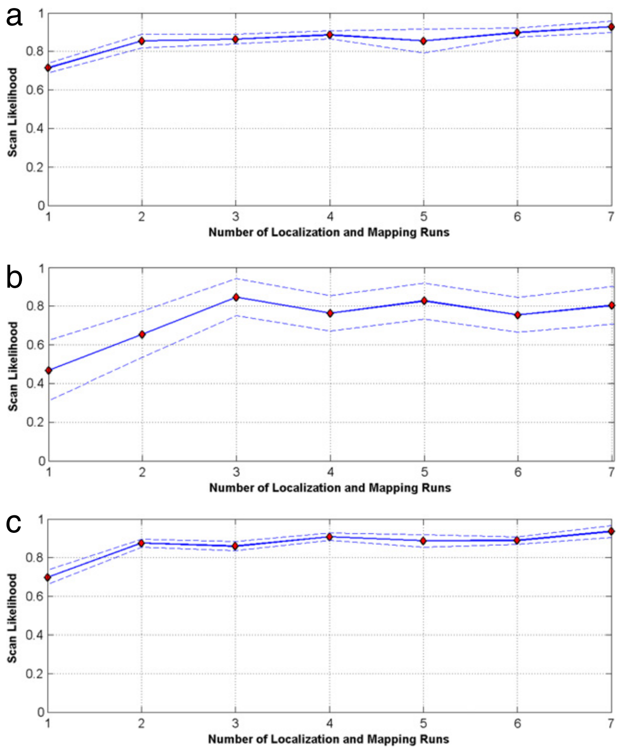

Fig. 13. Mean scan likelihood over the mapping and localization runs. (a) Ground floor. (b) First floor. (c) Second floor.

which is greater than the LRF maximum range (4 m). In addition, the first level is traversed by many pedestrians. Even though the SLAM algorithm presented slight divergences over the mapping and localization runs, they were not catastrophic enough to get the robot lost.

It is worth noting that, the increasingly improved behavior of the scan likelihood over the mapping and localization runs means that the localization algorithm behaves well using the persistent configuration of the environment. In addition, the LTM map of the environment (including vertical edge features and laser scans) is changed properly as the configuration of the environment is modified over time.

5.3.2. Mean pose error along the robot trajectory

Fig. 14 shows the mean pose (*XY* and heading) error along the robot trajectory for the ground (Fig. 14(a) and (b)), first (Fig. 14(c) and (d)), and second floor (Fig. 14(e) and (f)). The procedure to obtain the pose error was as follows: first, only one robot trajectory was considered. This trajectory corresponds to the first mapping and localization run. Second, this trajectory was evaluated (i.e. only the robot localization was considered) using all other maps corresponding to the seven mapping and localization runs. Third, in order to obtain the ground truth data, we used the G^2O framework [47]. This algorithm provides a solution for batch optimization of graph-based non-linear error functions. We introduced the first robot trajectory of each experiment as a graph, including nodes, edges and constraints. The first mapping and localization run of each experiment was carefully selected for ground-truth estimation purposes. The output of the algorithm is an estimate of the robot trajectory obtained after a non-linear minimization using a Levenberg–Marquardt algorithm. Finally, using this output we were able to extract the *X*–*Y* and heading error of the robot along its path as depicted in Fig. 14. The aim of this test is to show the robot mean pose error behaves well despite of changing illumination conditions, dealing with dynamic objects and occlusions.

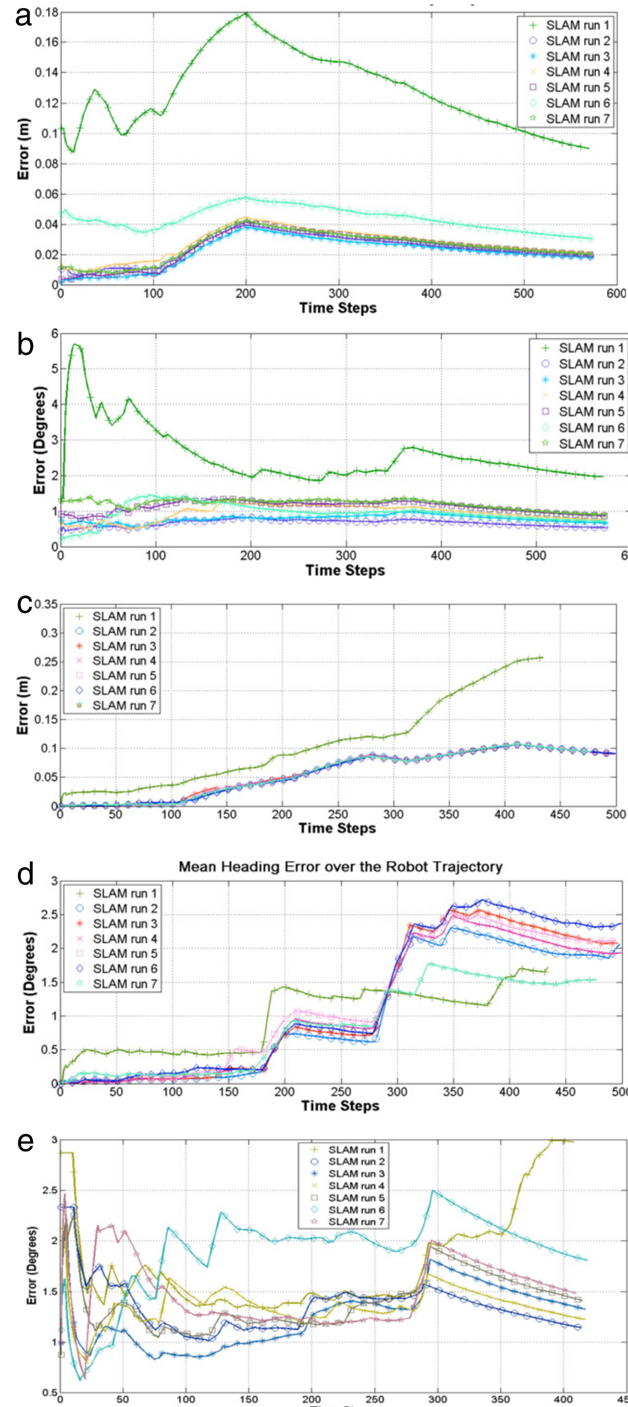

Fig. 14. Mean pose error along the robot trajectory. (a)–(b) *XY* and heading error for the ground floor. (c)–(d) *XY* and heading error for the first floor. (e)–(f) *XY* and heading error for the second floor.

In general, comparing with the first mapping and localization run, the *XY* and heading error graphs depicted in Fig. 14 decrease as more are performed. This can be directly observed in those figures corresponding to the ground and second floor. It is worth noting that the rise and fall of the *XY* and heading error graphs depend on the environmental conditions, which include situations such as pedestrians passing by at some particular node in the map, changes in illumination (time of day or season), occlusions, and changes in furniture location. Nevertheless, whatever the changes, a decreasing behavior was evident, which followed the peak in error graph. Fig. 13(c) and (d) shows the *XY* and

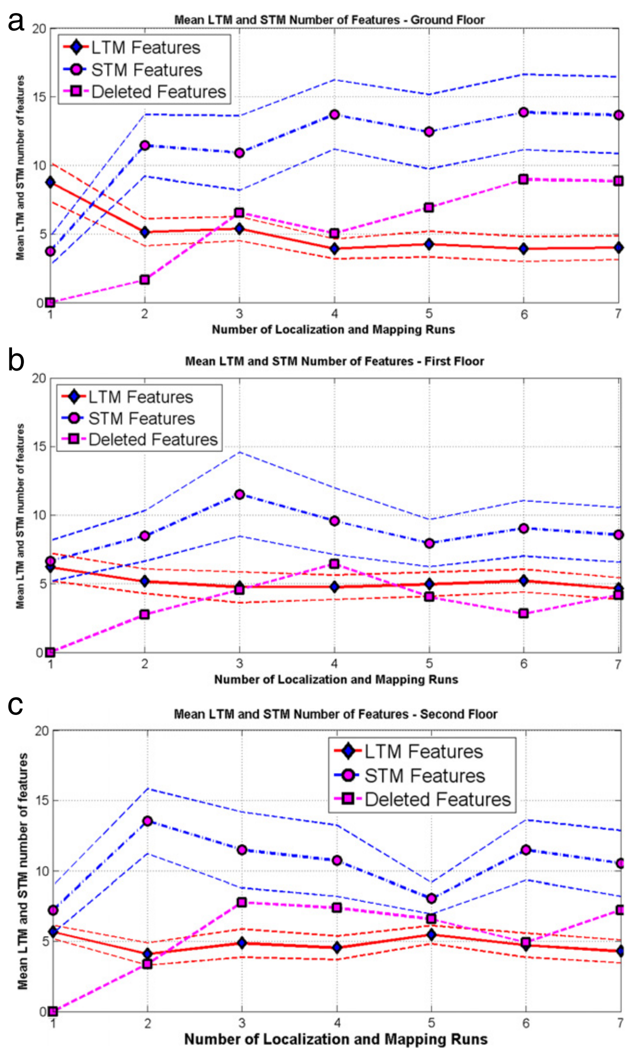

Fig. 15. Mean LTM, STM and deleted number of features by node over the mapping and localization runs. (a) Ground floor. (b) First floor. (c) Second floor.

heading error corresponding to the first floor, which is a very busy floor and where the SLAM algorithm used encountered more difficulties due the lack of availability of features over 6 m (exceeding the maximum range of the LRF—4 m). However, since the FastSLAM algorithm performs a non-Gaussian joint posterior distribution modeling of the robot pose, and designates the feature classification performed by the FSH model as either LTM or STM, the slight divergences were not catastrophic. This situation can be also observed in Fig. 14(b), where the scan likelihood uncertainty is greater than the scan likelihood uncertainty in Fig. 12(a) and (c).

5.3.3. Scalability

An important motivation behind this work is to deal with large environments and long-term navigation. Thus, the mean number of LTM, STM, and deleted features per node provide evidence for the scalability of this work. Fig. 15 shows the evolution of the number of LTM, STM, and deleted features by node as the mapping and localization runs increase. The diamond points correspond to the evolution of the number of LTM features, the circle points are the evolution of the number of STM features, and the square points are the evolution of the number of STM deleted features. The dashed curves show their uncertainties.

Observing the evolution of LTM and STM features in these figures, there is a clear tendency for LTM features to remain almost constant. To start with, the number of LTM features is greater than the number of LTM features in the other mapping and localization runs. An explanation of this behavior is that at the very beginning most of the features are considered to be LTM, however as more mapping and localization runs are performed the number of LTM features tends to decrease, because the more stable vertical edges are identified. On the other hand, as expected, the number of STM features is greater than LTM features, but thanks to the pruning method discussed in Section 3.2.2.2 the number of STM features does not increase boundlessly.

5.3.4. Matching effort

Classifying the environmental features as STM or LTM has another interesting result it reduces the mean matching effort in comparison with the full matching effort. In this work, the matching effort is the mean percentage of LTM or STM matched features over the mapping and localization runs. Fig. 16 shows the mean matching effort for the map on the ground, first, and second floor over the mapping and localization runs when the FSH model was used and when it was not. In this figure, diamond points correspond to the LTM matching effort over the mapping and localization runs, square points are the STM matching effort, and circle points are the full matching effort (without using the FSH model).

Observing Fig. 16, the LTM matching effort is greater than the STM matching effort because the more stable the (LTM) features the more likely they are to be found compared with STM features. This result, in conjunction with the scan likelihood depicted in Fig. 13, shows evidence that the FSH model proposed in this work updates the map of the environment in accordance with the changes that have taken place on it, and it also shows that the FSH model includes these changes to increase the localization accuracy.

The full matching effort was measured without using the FSH model, that is, it means using all the features available, without classifying them as STM or LTM, and without pruning the useless or old STM features. Fig. 15(a)–(c) show the full matching effort is greater than the LTM and STM matching effort. As a result, reducing the number of matching candidates also reduces the data association effort for long-term runs, and increases the robustness in dynamic environments reducing the effect of outliers.

6. Conclusions

This paper presented a more complete Feature Stability Histogram model, one able to be used with current SLAM methods, and thanks to the feature classification and removal methods proposed it can deal with long-term mapping and localization runs in dynamic environments. Further to our previously published results [7], this paper integrates the FSH model with the FastSLAM algorithm and it presents a wide experimental analysis in long-term mapping and localization experiments using an indoor dataset captured over one year at different time of day and seasons of the year. In addition, the environment appearance was modeled using vertical edges and their positions were estimated from the sensor model proposed in [42].

The FSH model is an innovative feature management approach that is inspired by the human memory model [22], implementing concepts such as LTM and STM as mechanisms to classify features as either stable or non-stable, and removing useless and old features, avoiding in this way the ever-increasing number of features that would cause problems in long-term mapping and localization runs. The proposed feature classification method as LTM or STM is based on the *k-means* algorithm and takes advantage of feature strength. Furthermore, using this clustering algorithm, weak and old STM features were detected and deleted by considering the normalized feature time stamp.

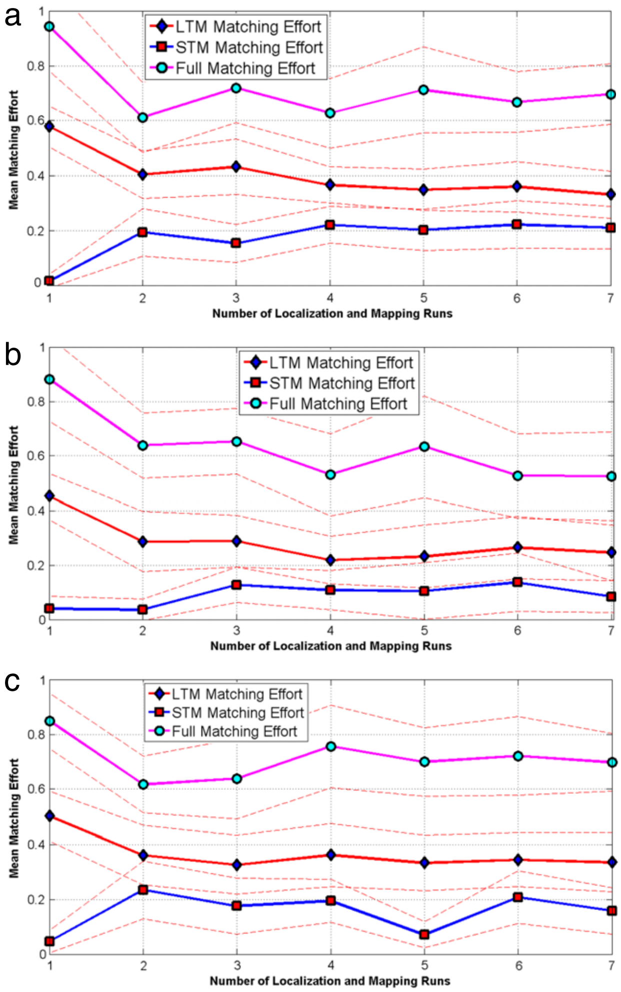

Fig. 16. Matching effort over the mapping and localization runs with the FSH model and without the FSH model. (a) Ground floor. (b) First floor. (c) Second floor.

The FSH model proposed in this paper implements these two ideas to deal with long-term mapping and localization in dynamic environments. As a result, qualitative and quantitative results were achieved. Qualitative results include, first, the dynamic objects as pedestrians passing by were filtered out, avoiding robot position estimation errors, and then reducing laser alignment errors, which tend to generate thick and blurred walls. Dynamic objects also cause spurious features, though they were filtered out and they do not appear in the LTM map which was used for robot localization. Second, the map quality increases over time, as, the more stable features are continuously rehearsed and the more stable changes in the environment are integrated into the LTM map. The more mapping and localization runs performed, the more consistent the vertical edges with the appearance of the environment when the FSH model is used. Finally, updating the map in accordance with the changes observed in the environment was also demonstrated. The LTM map of the environment embeds the changes in the environment, which means that not only is the configuration of the environment learned, but corrections in the map of the environment were also performed.

In the absence of ground-truth, four performance measures were carried out: first, the laser scan likelihood over the mapping and localization runs has shown an increasingly improved behavior, meaning that the localization accuracy also increased when the FSH model is used. In addition, as the environment changes, the LTM map of the environment changes accordingly. Second, the mean pose error along the robot trajectory has shown how the *XY* and heading error evolves over the robot path and the mapping and localization runs. In general, the more mapping and localization runs performed, the less the error poses. However, depending on the environmental conditions error peaks were present but followed by a fall in the error graph. Third, the mean number of LTM, STM and deleted features by node provide evidence for the scalability of this work. In this way, the results reported here show that the FSH model integrated in a SLAM solution deals well with large environments because LTM features were only used for robot mapping and localization, and useless or old STM features were deleted properly. Finally, the LTM, STM, and full matching efforts provide evidence of reducing the effect of outliers. Also, taking into account the scan likelihood results, the localization accuracy is increased when the FSH model was used, because it incorporates the current stable changes of the environment into the LTM map.

As for future work, better landmark rating methods are needed because the feature covariance suffers from the well-known disadvantage of obtaining an overconfident uncertainty as the map is continuously updated. It would be interesting to fuse feature visual appearance and its metric information. The experiments reported here and in our previous published results [7] correspond to indoor environments, so conducting outdoor experiments would be an interesting option to evaluate the FSH model. However, this could be difficult because of the lack of availability of long-term outdoor datasets. Another interesting question is related with knowing the feature pruning decision is right or wrong, and what are the consequences of wrong feature pruning on map consistency or robot pose.

In conclusion, the main advantage of the proposed FSH model is the manner in which it incorporates changes and extracts features from the environment configuration, incrementally increasing map quality. This result arises from the fact that rating map landmarks allows them to be classified as STM or LTM in the context of the modified human memory model [7,22]. As a result, the localization accuracy is increased in further mapping and localization runs, and the data association effort is reduced thanks to the map landmark classification approach into STM and LTM, and the pruning method described in this paper. Therefore, integrating the proposed FSH model into a SLAM method improves its behavior in long-term navigation.

Acknowledgments

This work has been partially supported by the project RAIMON—Autonomous Underwater Robot for Marine Fish Farms Inspection and Monitoring (Ref. CTM2011-29691-C02-02) funded by the Spanish Ministry of Science and Innovation, the LASPAU-COLCIENCIAS grant 136-2008, the University of Valle contract 644-19-04-95, and the consolidated research group's grant SGR2009-00380.

References

[1] R. Qiu, Z. Ji, A. Noyvirt, A. Soroka, R. Setchi, D. Pham, S. Xu, N. Shivarov, L. Pigini, G. Arbeiter, F. Weisshardt, B. Graf, M. Mast, L. Blasi, D. Facal, M. Rooker, R. Lopez, D. Li, B. Liu, G. Kronreif, P. Smrz, Towards robust personal assistant robots: experience gained in the SRS project, in: Intelligent Robots and Systems, IROS, vol. 1, 2012, pp. 1651–1657. (1).
[2] P. Pfaff, Graph-based SLAM in industrial environments, in: KUKA Laboratories GmbH, Workshop for 2011 Robotics: Science and Systems Conference, vol. 1, 2011, pp. 1–8. (1).
[3] M. Robots, PatrolBot robot platform. Web page http://www.mobilerobots.com/researchrobots/researchpatrolbot.aspx (last seen April, 2013).
[4] P. Ridao, M. Carreras, D. Ribas, R. Garcia, Visual inspection of hydroelectric dams using an autonomous underwater vehicle, Journal of Field Robotics 1 (27) (2010) 759–778.
[5] T.M. Howard, A. Morfopoulos, J. Morrison, Y. Kuwata, C. Villalpando, L. Matthies, M. McHenry, Enabling continuous planetary rover navigation through FPGA stereo and visual odometry, in: 2012 IEEE Aerospace Conference, 3–10 March 2012, pp. 1–9.

[6] T. Bailey, H. Durrant-Whyte, Simultaneous localization and mapping (SLAM): part II, Robotics and Automation Magazine, IEEE 13 (2008) 108–117.
[7] B. Bacca, J. Salvi, X. Cufí, Appearance-based mapping and localization for mobile robots using a feature stability histogram, Robotics and Autonomous Systems 59 (2011) 840–857.
[8] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges, in: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI, 2003, pp. 1151–1156.
[9] B. Bacca, E. Mouaddib, X. Cuffi, Embedding range information on omnidirectional images through laser range finder, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS, vol. 1, 2010, pp. 2053–2058.
[10] C. Wang, C. Thorpe, S. Thrun, Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas, in: Proceedings. ICRA IEEE International Conference on Robotics and Automation, vol. 1, 2003, pp. 842–849.
[11] W. Burgard, C. Stachniss, D. Haehnel, Mobile Robot Map Learning From Range Data in Dynamic Environments, in: Autonomous Navigation in Dynamic Environments, vol. 35, Springer-Verlag, 2007, pp. 3–28.
[12] J. Andrade-Cetto, A. Sanfeliu, Concurrent map building and localization on indoor dynamic environments, International Journal of Pattern Recognition and Artificial Intelligence 16 (2002) 361–374.
[13] S. Hochdorfer, C. Schlegel, Landmark rating and selection according to localization coverage: addressing the challenge of lifelong operation of SLAM in service robots, in: IROS IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, 2009, pp. 382–387.
[14] K. Pirker, M. Ruther, H. Bischof, CD SLAM—continuous localization and mapping in a dynamic world, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, vol. 1, 2011, pp. 3990–3997.
[15] H. Kretzschmar, G. Grisetti, C. Stachniss, Life-long map learning for graph-based SLAM approaches in static environments, KI Künstliche Intelligenz 24 (2010) 1–7.
[16] K. Konolige, J. Bowman, Towards lifelong visual maps, in: IROS IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, 2009, pp. 1156–1163.
[17] K. Konolige, J. Bowman, J.D. Chen, P. Mihelich, M. Calonder, V. Lepetit, P. Fua, View-based maps, The International Journal of Robotics Research 30 (2011) 1728–1754.
[18] P. Biber, T. Duckett, Experimental analysis of sample-based maps for long-term SLAM, The International Journal of Robotics Research 28 (2009) 20–33.
[19] M. Milford, G. Wyeth, Persistent navigation and mapping using a biologically inspired SLAM system, The International Journal of Robotics Research 29 (2009) 1131–1153.
[20] A.J. Glover, W.P. Maddern, M. Milford, G. Wyeth, FAB-MAP + RatSLAM: appearance-based SLAM for multiple times of day, in: IEEE Int. Conf. on Robotics and Automation, ICRA, vol. 1, 2010, pp. 3507–3512.
[21] M. Cummins, P. Newman, Highly scalable appearance-only SLAM—FAB-MAP 2.0, in: Proceedings of Robotics: Science and Systems, vol. 1, 2009, pp. 1–8.
[22] R.C. Atkinson, R.M. Shiffrin, Human memory: a proposed system and its control processes, in: The Psychology of Learning and Motivation, Academic Press, New York, 1968, pp. 89–195.
[23] A. Baddeley, Working memory: looking back and looking forward, Nature Reviews Neuroscience 4 (2003) 829–839.
[24] F. Dayoub, G. Cielniak, T. Duckett, Long-term experiments with an adaptive spherical view representation for navigation in changing environments, Robotics and Autonomous Systems 59 (2011) 285–295.
[25] M. Labbe, F. Michaud, Memory management for real-time appearance-based loop closure detection, in:IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, vol. 1, 2011, pp. 1271–1276.
[26] R. Llinas, I of the Vortex: From Neurons to Self, The MIT Press, 2002.
[27] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Computer Vision and Image Understanding 110 (2008) 346–359.
[28] S. Lloyd, Least squares quantization in PCM, IEEE Transactions on Information Theory 28 (1982) 129–137.
[29] G. Dissanayake, H.F. Durrant-Whyte, T. Bailey, A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem, in: Proceedings. ICRA IEEE International Conference on Robotics and Automation, vol. 1, 2000, pp. 1009–1014.
[30] K.V. Mardia, J.T. Kent, J.M. Bibby, Multivariate Analysis (Probability and Mathematical Statistics), Academic Press, 1980.
[31] J.C. Lagarias, J.A. Reeds, M.H. Wright, P.E. Wright, Convergence properties of the Nelder–Mead simplex method in low dimensions, SIAM Journal on Optimization 9 (1) (1998) 112–147.
[32] A. Hokuyo, Scanning range finder URG-04LX. [Online] Available at: http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html (accessed 2012).
[33] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, J. Tardós, A comparison of loop closing techniques in monocular SLAM, Robotics and Autonomous Systems 57 (12) (2008) 1188–1197.
[34] L. Clemente, A. Davison, I. Reid, J. Neira, J.D. Tardós, Mappinglarge loops with a single hand-held camera, in: Proc. Robotics: Science and Systems Conference, vol. 1, 2007, pp. 1–8.
[35] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, J. Tardos, An image-to-map loop closing method for monocular SLAM, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, 2008, pp. 2053–2059. (1).
[36] G.D. Tipaldi, K.O. Arras, FLIRT—Interest regions for 2D range data, in: Robotics and Automation, ICRA, vol. 1, 2010, pp. 3616–3622. (1).
[37] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), The MIT Press, 2005.
[38] W. Rucklidge, Efficient Visual Recognition Using the Hausdorff Distance, Springer-Verlag New York, Inc., 1996.
[39] C. Estrada, J. Neira, J.D. Tardos, Hierarchical SLAM: real-time accurate mapping of large environments, IEEE Transactions on Robotics 21 (2005) 588–596.
[40] J. Aulinas, X. Lladó, J. Salvi, Y. Petillot, Selective submap joining for underwater large scale 6-DOF SLAM, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS, vol. 1, 2010, pp. 2552–2557.
[41] G. Grisetti, D.L. Rizzini, C. Stachniss, E. Olson, W. Burgard, Online constraint network optimization for efficient maximum likelihood map learning, in:IEEE International Conference on Robotics and Automation, ICRA, vol. 1, 2008, pp. 1880–1885.
[42] B. Bacca, J. Salvi, X. Cufi, Vertical edge-based mapping using range-augmented omnidirectional vision sensor, IET-Computer Vision 7 (2) (2013) 135–143.
[43] J. Bazin, C. Demonceaux, P. Vasseur, Fast central catadioptric line extraction, in: Pattern Recognition and Image Analysis, in: Lecture Notes in Computer Science, vol. 4478, Springer, Berlin, Heidelberg, 2007, pp. 25–32.
[44] J.P. Barreto, H. Araujo, Geometric properties of central catadioptric line images and their application in calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1327–1333.
[45] J. Neira, J.D. Tardos, Data association in stochastic mapping using the joint compatibility test, IEEE Transactions on Robotics and Automation 17 (2001) 890–897.
[46] A. Pronobis, B. Caputo, COLD: COsy localization database, The International Journal of Robotics Research (IJRR) 28 (2009) 588–594.
[47] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, g^2O: a general framework for graph optimization, in: IEEE International Conference on Robotics and Automation, ICRA, vol. 1, 2011, pp. 3607–3613.

Bladimir Bacca Cortes graduated in Electronic Engineering in 1999, he received his M.Sc. in 2004 both at the Universidad del Valle, Cali, Colombia; he got his Ph.D. at the University of Girona. He is a Professor of the Electrical and Electronic Engineering School at the Perception and Intelligent Systems Group, Universidad del Valle. His current research interests are in the field of localization and mapping for mobile robots, computer vision, focusing on SLAM and appearance-based environmental models.

Joaquim Salvi graduated in Computer Science at the Technical University of Catalonia in 1993, received the DEA (M.Sc.) in Computer Science in 1996 and the Ph.D. in Industrial Engineering in 1998 both at the University of Girona, Spain. He is a Professor of Computer Vision at the Computer Architecture and Technology Department and at the Computer Vision and Robotics Group, University of Girona. His current interests are in the field of computer vision and mobile robotics, focused on visual SLAM, structured light, stereo vision, and camera calibration. He is the leader of the 3D Perception Lab and charter member of the spinoffs AQSense and BonesNotes. Dr. Salvi received the Best Thesis Award in engineering for his Ph.D.

Xavier Cufí received a B.S. in Physic Sciences from the Universitat Autònoma de Barcelona (UAB) in 1987, and Ph.D. in Computer Vision from the Universitat Politènica de Catalunya (UPC) in 1998. He is a lecturer and the academic director of the Computer Engineering department at the Universitat de Girona (UdG), and he belongs to the Computer Vision and Robotics (VICOROB) research group of the UdG. He is also involved on the Academic Board of an International Erasmus-Mundus master on Vision and Robotics. His recent research interests include underwater image processing, underwater robotics, and mobile robotics. Dr. X. Cufí is a member of the Spanish Committee on Automatics (CEA—GTRobotics) and Spanish member of the International Federation of Automatic Control (IFAC).