

Concurrent Map Building and Localization on Indoor Dynamic Environments

Juan Andrade-Cetto and Alberto Sanfeliu
Institut de Robòtica i Informàtica Industrial, UPC-CSIC
Llorens i Artigas 4-6, Edifici U, 2a pl, Barcelona 08028, Spain
cetto@iri.upc.es

Abstract

A system that builds and maintains a dynamic map for a mobile robot is presented. A learning rule associated to each observed landmark is used to compute its robustness. The position of the robot during map construction is estimated by combining sensor readings, motion commands, and the current map state by means of an Extended Kalman Filter. The combination of landmark strength validation and Kalman filtering for map updating and robot position estimation allows for robust learning of moderately dynamic indoor environments.

Keywords: Mobile robot map learning, mobile robot navigation, CML, SLAM, Extended Kalman Filter.

1 Introduction

Efficient indoor mobile robot navigation is limited mainly by the ability of a robot to perceive and interact with its surroundings in a deliberative way. And, for such interaction to take place, a model or description of the environment usually needs to be specified beforehand. If a global description or measurement of the elements present in the environment is not available, the descriptors and methods for the autonomous building of one are required. This is, either the robot has a global map, or it is given the means to learn one.

Many systems that incorporate human-made models of the environment have been successfully developed, even when only an approximate map is given or when the system must navigate in crowded environments^{12,18}. However, the autonomous building of a global, and possibly dynamic, map of the environment for a mobile robot is still a difficult problem. Three main difficulties arise during autonomous learning of an indoor map by a mobile robot:

1. Dead reckoning. As the robot moves, its global position estimate from encoder readings accumulates drift errors due to wheel slippage or encoder quantization. And, after a small period of time, unless corrected, that estimate is unreliable.
2. Sensors. Obstacle and landmark position estimates are restricted by the type of sensors used, and by the finesse of the algorithms used for extraction, location, and identification. For example, in the case of sonar sensors, false reflections are typical, and a large amount of obstacle position readings might be inaccurate. On the other hand, if vision is used, effective and accurate extraction and identification of landmarks is often prohibitive for real-time systems.
3. Dynamic environments. In a restricted number of applications, the environment remains static. However, for the general case, obstacle locations usually change over time, pushing for stochastic map models that continuously update the environment map to reflect these changes.

Map construction in mobile robotics has been achieved typically by updating grid maps of obstacles. Recent contributions on grid-based map building include the ones by Anousaki², Duckett¹¹, Lee¹⁵, and Oriolo¹⁹ among others.

If a mobile robot is able to repeatedly identify the same landmark in the environment, it could refine the position of such landmark in the map, and at the same time, use this information to update the estimate it has about its own current location. To this aim, Castellanos⁶, Kwon¹⁴, and Dissanayake¹⁰ to name a few, have suggested probabilistic approaches for obstacle parameterization and robot localization. Another technique that uses a Bayesian approach to map learning and localization over grid maps is by Thrun²⁰.

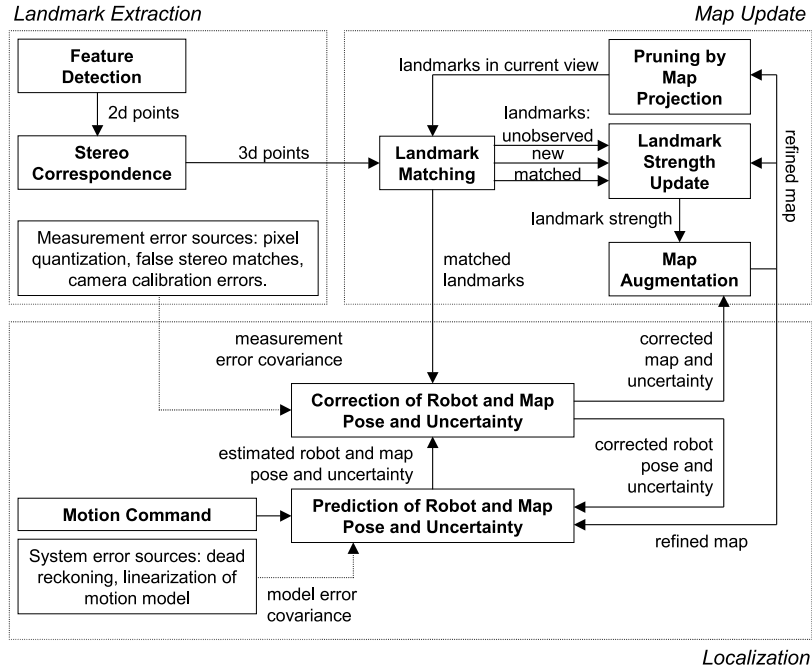


Figure 1: System architecture.

A completely different approach to robot navigation that does not take into account the geometric structure of the environment relies on the use of goal oriented cognitive maps to learn the relationship between successively explored places^{3,4,7,13}. However, all of the above cited contributions are limited in that changing environments are only dealt at most with reactive behaviors.

In this article we present a methodology for the construction and update of a dynamic map for a mobile robot. It extends on previous probabilistic methods to concurrent localization and mapping by incorporating the measurement of landmark strength and quality, measures that permit the elimination of unreliable observations so that map updating can occur even in moderately changing environments. Unlike grid-based techniques, it is scale independent. The technique does not make any assumption on the distribution of the landmark positions, but it does expect white distribution of dead reckoning and sensor measurement errors. Our system architecture is shown in Fig. 1. It includes three distinctive modules: landmark extraction, map updating, and localization.

2 Landmark Strength and Quality Measures

The use of an estimation-theoretic approach such as Extended Kalman Filtering^{5,17,21} to concurrent localization and mapping has gain popularity in recent years^{1,6,10,16}. However, little attention has been paid on the constraints inherent with such formalism. Two of the most critical of these restrictions are the linearization constraint and the data association constraint. The former refers to the requisite that a linearized model for the motion of the mobile robot be available, as well as accurate models for the uncertainty in sensor measurements and motion commands. The latter constraint refers to the issue of landmark identification, and consequently, to the issue of landmark quality.

In the work by Dissanayake¹⁰ for example, landmark robustness is addressed as an implementation issue only, suggesting a quality measure based on the probability density function of the observations associated to a given landmark, disregarding the temporal dispersion of such observations.

We introduce in this communication a set of measures for landmark validation. Measures that quantify how disperse both in space and time landmark observations are in order to be considered strong references for environment representation and robot localization. Temporary landmarks and those coming from noisy sensor readings are pruned from the map as their strength measure diminishes over time. By the same token, those landmarks that are repeatedly observed are considered trusted indicators of the structure of the environment. Similarly, those landmarks that accumulate wide spatial dispersion during map building are also removed.

2.1 Temporal Uncertainty

In order to evaluate landmark temporal dispersion, we have resorted to an exponential decay rule with the possibility to link neighboring landmarks in a networked representation.

Consider the map of the environment to be a set of landmarks parameterized as quadruples

$$T = \{(\mathbf{z}_i^W, \Sigma_i^W, \zeta_i^W, \mathcal{X}_i), 0 \leq i \leq t\}, \quad (1)$$

with \mathbf{z}^W the landmark position in world coordinates, Σ_i^W the associated landmark covariance matrix, ζ_i^W a vector of landmark appearance properties, \mathcal{X}_i the strength state, and t the total number of landmarks in the map.

For each landmark in T , there exists an associated memory cell that will register how persistent, and how old the landmark is. The state of the cell $\mathcal{X}_i(k)$ will be the strength state of landmark i , and $\bar{e}_i = \{0, 1\}$ its identification

stamp. The proposed update rule equation is

$$\mathcal{X}_i(k) = \frac{1}{1 + e^{-(\alpha_i \bar{e}_i + \beta_i \mathcal{X}_i(k-1) + \sum_{j \in V(k)} w_{ji}(k-1) \mathcal{X}_j(k-1))}} \quad (2)$$

with the long-term learning expression

$$w_{ji}(k) = w_{ji}(k-1) + \gamma \mathcal{X}_i(k-1) \mathcal{X}_j(k-1) \quad (3)$$

where α_i is an input weight used to regulate the contribution of the identification of a landmark in the current scene over the previous map configuration; β_i is a memory weight used to regulate the contribution of the previous state of a given landmark over its new state; and γ is a positive constant that determines the speed of learning. The sum of weights into a state node i is normalized to restrict them from growing indefinitely over time. As will be seen later, the strength states are only updated for those landmarks that fall on a projected view of the map over the robot field of view.

2.2 Spatial Uncertainty

It has been shown¹⁰ that the a posteriori error covariance estimate

$$\mathbf{P}(k|k) = E[(\mathbf{x}(k) - \mathbf{x}(k|k))(\mathbf{x}(k) - \mathbf{x}(k|k))^\top] \quad (4)$$

for any landmark in the map state estimate of the Kalman filter decreases monotonically, and that it converges to a lower bound at the end of the map building process. The term $(\mathbf{x}(k) - \mathbf{x}(k|k))$ refers to the error between the true robot and landmark locations and their a posteriori estimate. However, by maintaining an unbiased sample covariance of the observations we can still compute a measure for the spatial distribution of that given landmark. Given an observation $(\mathbf{z}_i, \mathbf{0}, \zeta_i, 1)$ and its corresponding projected match $(\mathbf{z}_j, \Sigma_j, \zeta_j, \mathcal{X}_j)$, the running unbiased sample landmark covariance matrix update can be computed with

$$\Sigma_j(k) = \frac{(n_j - 2)\Sigma_j(k-1) + (\mathbf{z}_j(k) - \mathbf{z}_i(k))(\mathbf{z}_j(k) - \mathbf{z}_i(k))^\top}{n_j - 1} \quad (5)$$

with n_j a landmark identification accumulator that is incremented if a scene to map match is obtained for such landmark z_j .

In the following sections we show how the landmark position estimates in the robot reference frame, their appearance properties, and the correlation information for commonly persistent landmarks in the scene, will help in building the map, as well as to position the robot within this map minimizing the localization error.

3 Map Update

Map building is performed iteratively during robot navigation. The result from the landmark extraction module is a set of detected landmark positions in robot coordinates and their corresponding appearance property vectors

$$D(k) = \{(\mathbf{z}_1, \zeta_1), (\mathbf{z}_2, \zeta_2), \dots, (\mathbf{z}_d, \zeta_d)\}. \quad (6)$$

The previous map state is as in Eq. 1 the set of quadruples

$$T(k-1) = \{(\mathbf{z}_i^W, \Sigma_i^W, \zeta_i^W, \mathcal{X}_i), 0 \leq i \leq t\}, \quad (7)$$

with $T(0) = \{\emptyset\}$.

3.1 Pruning of Possible Matches by Map-to-Scene Projection

The first step in the construction and update of the map is the search for landmark matches between $D(k)$ and $T(k-1)$. In order to make the system computationally efficient, this search must be limited to a reduced number of landmarks $V(k)$. The elements of $V(k)$ are computed by projecting $T(k-1)$ onto the scene and trimmed to those landmarks that fall within the current robot field of view.

The position of the robot in world coordinates is $\mathbf{x}_r = [x, y, \theta]^\top$, with $\mathbf{R} = \text{Rot}(z, \theta)$ and $\mathbf{t} = [x, y, 0]^\top$. A state vector that encompasses both the robot pose and the location of each landmark in world-centered coordinates will be updated during the localization step of the algorithm

$$\mathbf{x}(k) = \left[\mathbf{x}_r(k)^\top, \mathbf{z}_1^{W^\top}, \mathbf{z}_2^{W^\top}, \dots, \mathbf{z}_t^{W^\top} \right]^\top \quad (8)$$

The function that changes a landmark in the map from world coordinates to robot coordinates is given by

$$\psi : (\mathbf{z}^W, \Sigma^W, \zeta^W, \mathcal{X}) \mapsto (\mathbf{z}, \Sigma, \zeta, \mathcal{X}) = (\mathbf{R}^\top (\mathbf{z}^W - \mathbf{t}), \mathbf{R} \Sigma^W \mathbf{R}^\top, f(\zeta^W, \mathbf{x}_r), \mathcal{X}). \quad (9)$$

$V(k) \subseteq \psi(T(k-1))$, and the set inclusion rules for $V(k)$ are

$$0 \leq u \leq 2u_0, \quad 0 \leq v \leq 2v_0, \quad z_{\min} \leq z \leq z_{\max} \quad (10)$$

where the function $f(\cdot)$ is used to represent appearance properties in robot centered coordinates, (u, v) are the coordinates of the projection of \mathbf{z} onto a robot mounted camera image plane, (u_0, v_0) are the image center coordinates, and (z_{\min}, z_{\max}) is a user defined depth threshold. A schematic representation of these inclusion areas for each instance k is depicted as triangular regions in Fig 2.

3.2 Scene-to-Map Landmark Matching

If the coordinates of a sensed landmark fall within a distance d_{\max} of a landmark in the pruned set $V(k)$, and provided their appearance vectors are similar, then we have a scene-to-map landmark match. By rewriting the sets $D(k)$ and $V(k)$ in the forms

$$D(k) = \{(\mathbf{z}_i, \mathbf{0}, \zeta_i, 1), 0 \leq i \leq d\}, \quad V(k) = \{(\mathbf{z}_j, \Sigma_j, \zeta_j, \mathcal{X}_j), 0 \leq j \leq v\}, \quad (11)$$

and provided a suitable appearance similarity metric $s(\cdot)$, the scene-to-map matching rules can be written as

$$(\mathbf{z}_i - \mathbf{z}_j)^\top \Sigma_j^{-1} (\mathbf{z}_i - \mathbf{z}_j) \leq d_{\max}, \quad s(\zeta_i, \zeta_j) \leq s_{\max}. \quad (12)$$

The use of the Mahalanobis distance in Eq.12, allow the computation of a suitable value for d_{\max} such that the null hypothesis that \mathbf{z}_i and \mathbf{z}_j match is not rejected at some desired confidence level. Nevertheless, the formulation of a sufficiently invariant similarity metric $s(\cdot)$ for the second matching criteria is a difficult task. The search for robust measures of appearance similarity as well as a world-to-robot appearance mapping function $f(\cdot)$ are active research topics in computer vision and are out of the scope of this paper. The choice of $s(\cdot)$ and $f(\cdot)$ for the experiments presented in this article are detailed in the implementation section.

The pairings of scene and map landmarks that satisfy the rules in Eq. 12 form a set of matches $M(k)$. The rest of the elements in $D(k)$ form a set $N(k)$ of new landmarks that must be added to the map. Also, there might exist a set $U(k)$ of unobserved landmarks due to occlusions, because these entities are no longer present in the scene, or because the illumination conditions do not allow the robot to identify them properly. The three sets M , N , and U make the set $U \cup V$.

$$M(k) = D(k) \cap V(k), \quad N(k) = D(k) - M(k), \quad U(k) = V(k) - M(k) \quad (13)$$

For an exemplary representation of the sets D, V, M, N , and U , see Fig. 2. The next step in map updating is to refine the strength of both the matched and unobserved sets of landmarks accordingly.

3.3 Map Augmentation

Once the strength and sample covariance have been updated for all landmarks in $V(k)$ by means of Eqs. 2-3,5, the next step is to augment (or trim) the map accordingly. As a first step, the landmarks in the map that meet the following two conditions

$$\mathcal{X}_j(k) < t_s, \quad \text{tr}(\Sigma_j(k)) > t_q \quad (14)$$

Time step k	Detected landmarks D	Pruned Map P	Matched landmarks M	New landmarks N	Unobserved Landmarks U	Forgotten landmarks F	Total map T
0	1,2,3			1,2,3			1,2,3
1	2,4,5	2	2	4,5			1,2,3,4,5
2	4,6,7	4,5	4	6,7	5	5	1,2,3,4,6,7

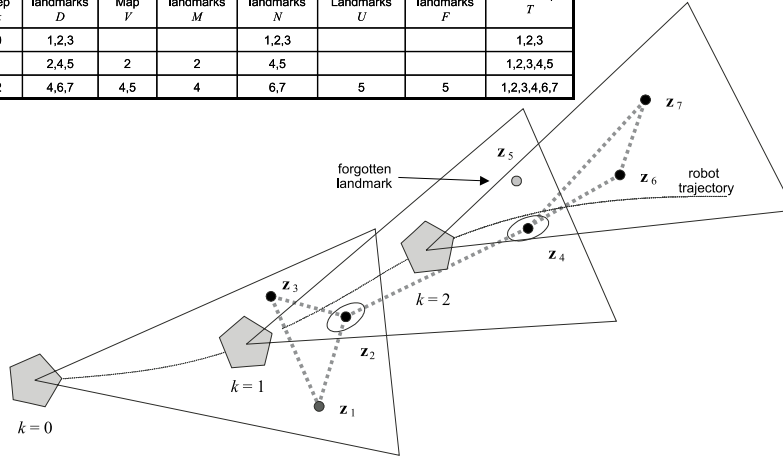


Figure 2: Concurrent localization and map building

are considered weak and must be removed from the map, with t_s and t_q user supplied landmark strength and quality thresholds. Consequently, their corresponding entries in the state vector \mathbf{x} are eliminated. For those landmarks that do not match the pruning conditions, and if $\mathbf{z}_j \in M(k)$, then the entries in $T(k)$ must be updated. After the computation of $\mathcal{X}_j(k)$ and $\Sigma_j(k)$, the state on the map for that landmark is computed with

$$T_j(k) = \psi^{-1}(\mathbf{x}_j, \mathbf{R}^\top \Sigma_j(k) \mathbf{R}, f^{-1}(\zeta_j, \mathbf{x}_r), \mathcal{X}_j(k)) \quad (15)$$

On the other hand, the newly observed landmarks $\mathbf{z}_j \in N(k)$ are initialized in the map with

$$T_j(k) = \psi^{-1}(\mathbf{z}_j, \mathbf{0}, f^{-1}(\zeta_j, \mathbf{x}_r), 1) \quad (16)$$

4 Localization

The equations used to update our estimate of the pose of the robot, and the location of the landmarks are next derived, provided a set of observations and the robot motion commands are available.

A nonlinear function that relates the measurements about the environment in Eq. 6 to the actual robot pose $\mathbf{x}_r(k)$, the landmark position estimates in the map $\mathbf{z}_i^W(k)$, and the uncertainty in sensor measurements $\mathbf{v}(k)$ at time step k is given by

$$\mathbf{z}_i(k) = \mathbf{h}_i(\mathbf{x}(k), \mathbf{v}(k)) \quad (17)$$

A noise-free approximation of Eq. 17 is given in Eq. 9 and rewritten here for clarity,

$$\mathbf{z}_i(k|k) = \mathbf{R}(k)^\top (\mathbf{z}_i^W - \mathbf{t}(k)) \quad (18)$$

with $\mathbf{R}(k) = \text{Rot}(\theta(k|k-1), z)$, and $\mathbf{t}(k) = [x(k|k-1), y(k|k-1), 0]^\top$. A linearized version of \mathbf{h}_i about this noise-free approximate can be formulated as a Taylor series with the higher order terms dropped, i.e.,

$$\mathbf{z}_i(k) \approx \mathbf{z}_i(k|k) + \frac{\partial \mathbf{h}_i(\mathbf{x}(k|k-1), 0)}{\partial \mathbf{x}} (\mathbf{x}(k) - \mathbf{x}(k|k-1)) + \mathbf{v}(k) \quad (19)$$

Differentiating Eq.17 with respect to the state vector $\mathbf{x}(k)$, we obtain the following expression for the Jacobian or measurement innovation matrix for landmark i at time step k :

$$\mathbf{H}_i(k) = [\mathbf{H}_r(k) \quad \mathbf{0} \quad \dots \quad \mathbf{0} \quad \mathbf{R}(k)^\top \quad \mathbf{0} \quad \dots \quad \mathbf{0}] \quad (20)$$

where

$$\mathbf{H}_r(k) = [\mathbf{0}_{3 \times 2} \mid \dot{\mathbf{R}}(k)^\top (\mathbf{z}_i^W(k) - \mathbf{t}(k))] - \mathbf{R}(k)^\top \quad (21)$$

Each iteration of an Extended Kalman Filter is divided into a prediction step (walking without sensing), and a correction step (refining from observations).

4.1 Prediction of Robot and Map Pose and Uncertainty

To predict the behavior of the system we need to add the motion command and system noise to the previous state estimation. The time update equations of the robot pose and map state and their error covariance matrix are

$$\mathbf{x}(k|k-1) = \mathbf{x}(k-1) + \mathbf{u}(k) \quad (22)$$

and

$$\mathbf{P}(k|k-1) = \mathbf{P}(k-1) + \mathbf{Q} \quad (23)$$

respectively, with $\mathbf{u}(k)$ the motion command vector, and \mathbf{Q} a given system noise matrix. \mathbf{Q} is typically set to a constant value, and can be computed by running a set of motion commands and parameterizing the deviation of the robot from the desired pose. If the number of samples is sufficiently large, white noise parameters can be expected for \mathbf{Q} .

4.2 Correction of Robot and Map Pose and Uncertainty

Given the prediction error $\mathbf{x}(k) - \mathbf{x}(k|k-1)$, the Kalman filter is designed to minimize the error covariance matrix in Eq. 4. During the correction step, the following two computations take place for each landmark in $M(k)$.

First, the Kalman Filter Gain is computed with

$$\mathbf{K}_i(k) = \mathbf{P}(k|k-1)\mathbf{H}_i^\top(k) (\mathbf{H}_i(k)\mathbf{P}(k|k-1)\mathbf{H}_i^\top(k) + \mathbf{R}^\top(k)\Sigma_i(k)\mathbf{R}(k))^{-1} \quad (24)$$

where $\Sigma_i(k)$ is the measurement covariance matrix at time k for landmark i computed in Eq. 5, and $\mathbf{R}^\top(k)\Sigma_i(k)\mathbf{R}(k)$ is the base change that will express this covariance matrix in robot centered coordinates, the reference frame where observations take place.

At this point, we are able to compute the robot pose estimate update from each observation,

$$\mathbf{x}(k) = \mathbf{x}(k|k-1) + \mathbf{K}_i(k)(\mathbf{z}_i - \mathbf{h}_i(\mathbf{x}(k|k-1), \mathbf{0})) \quad (25)$$

Eq. 25 shows how each of the observed landmarks contributes to the correction of the robot pose estimate $\mathbf{x}_r(k)$, and its own location $\mathbf{z}_i^W(k)$. Note that $\mathbf{z}_i^W(k)$ is the best approximation we have to the real landmark position in world coordinates at time step k , and that the measurement error $\mathbf{z}_i - \mathbf{h}_i(\mathbf{x}(k|k-1), \mathbf{0})$ is also an approximation to the actual error in locating that particular landmark. Given the convergence properties of the Kalman filter, and provided a sufficient number of measurements for each landmark are made over time, the error in the estimation of the robot pose and the landmark position will converge to a lower bound.

Similarly, the contribution to the error covariance estimate is obtained with

$$\mathbf{P}(k) = (\mathbf{I} - \mathbf{K}_i(k)\mathbf{H}_i(k)) \mathbf{P}(k|k-1) \quad (26)$$

Evaluating Eqs. 22-26 for all landmarks in $M(k)$ completes an iteration of the Kalman filter. The entire concurrent localization and map building algorithm is shown in Table 1.

5 Implementation Issues

For the implementation that supports the results presented in this article, the extraction of landmarks is based solely on visual information. A salient feature detector that uses Beaudet’s cornerness measure was implemented, with further refinement using the variance descent approach^{8,9}. These salient features are then pairwise matched in the stereo set by correlation, and by the enforcement of epipolar constraints. Each feature’s 3-d position with respect to the robot is reconstructed from stereo geometry. The 3-d position of a feature with respect to the robot \mathbf{z}_i , and an associated vector of appearance properties ζ_i , constitute a *landmark*. The appearance properties, which are used to validate scene to map landmark matches, include the pixel gray-level mean and distribution over a

for each iteration do:	
Landmark Extraction	
$D(k)$	← Feature detection and stereo correspondence.
Map Update	
$V(k)$	← Pruning of possible matches by map projection. <i>Input:</i> $T(k-1)$.
$M(k), U(k), N(k)$	← Scene to map landmark matching. <i>Input:</i> $D(k), V(k)$.
$\mathcal{X}(k)$	← Landmark strength update. <i>Input:</i> $M(k), N(k), U(k), T(k-1)$.
$T(k)$	← Map augmentation. <i>Input:</i> $T(k-1), \mathcal{X}(k), \mathbf{x}(k), \mathbf{P}(k)$.
Localization	
$\mathbf{x}(k k-1), \mathbf{P}(k k-1)$	← Map and robot pose prediction. <i>Input:</i> $\mathbf{x}(k-1), \mathbf{P}(k-1), \mathbf{u}(k), \mathbf{Q}$.
$\mathbf{x}(k), \mathbf{P}(k)$	← Map and robot pose correction. <i>Input:</i> $\mathbf{x}(k k-1), \mathbf{P}(k k-1), M(k)$.

Table 1: Concurrent localization and map building algorithm.

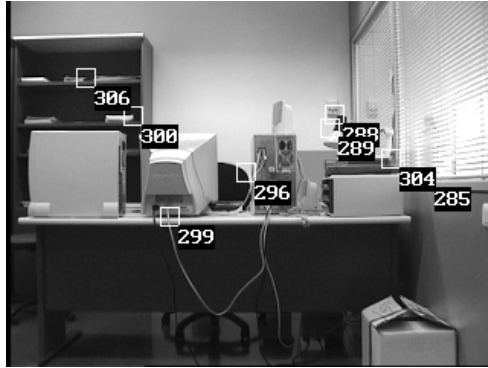
small window around the salient feature, and the energy of the feature computed from the corneriness measure.

The gray level appearance correlation around the pixels of interest weighted by the energy of the Beaudet corneriness measure is used as a similarity metric $s(\cdot)$, and the appearance mapping function $f(\cdot)$ is implemented by projecting the window of interest into world coordinates as in Eq. 9, and recomputing in this new window the appearance properties described before. Our test platform, the mobile robot MARCO, is shown in Fig. 3a, and a sample image of the feature extraction module is shown in Fig. 3b.

Fig. 4 shows a run of the full concurrent localization and map building algorithm with 50 landmarks detected. The localization of some of the landmarks might vary considerably due to the characteristics of the sensors. For example, if as in our case, computer vision is used, specular reflections might contribute to false readings. This is exemplified by the coarse localization of some landmarks in the figure with large uncertainty ellipses around them. By pruning the poor quality landmarks using the methodology described in this paper a more accurate map can be constructed, and consequently, better robot localization is achieved. The numbered dots in the figure correspond to the identified landmarks in the scene, and the surrounding ellipses indicate a scaled projection in the xy plane of their sample covariance Σ^W . The thin line square indicates the desired robot trajectory, and the thick line indicates the actual robot trajectory as estimated from landmark correspondences. Fig. 5 shows a three-dimensional view of the same run with the uncertainty ellipses projected in the room floor, and the landmark height indicated by thin vertical lines.



(a)



(b)

Figure 3: (a) Marco mobile robot. (b) Landmark extraction

The initial estimate of the robot pose in world coordinates must be known; e.g., $\mathbf{x}_r(0) = [0, 0, 0]^T$; and an initial estimate for the error covariance can be set to $\mathbf{P}(0) = \mathbf{I}$. Even if these initial parameters are not correct, the filter is guaranteed to converge to the actual robot pose; it will just take more time to do so.

6 Conclusions

The methodology for the concurrent localization and map building for a mobile robot was presented. Unlike grid-based techniques, it is scale independent. It was designed so that map updating can occur even in moderately changing environments by exploiting the relationship existing among neighboring landmarks, and the persistence of each landmark in the scene. It does not make any

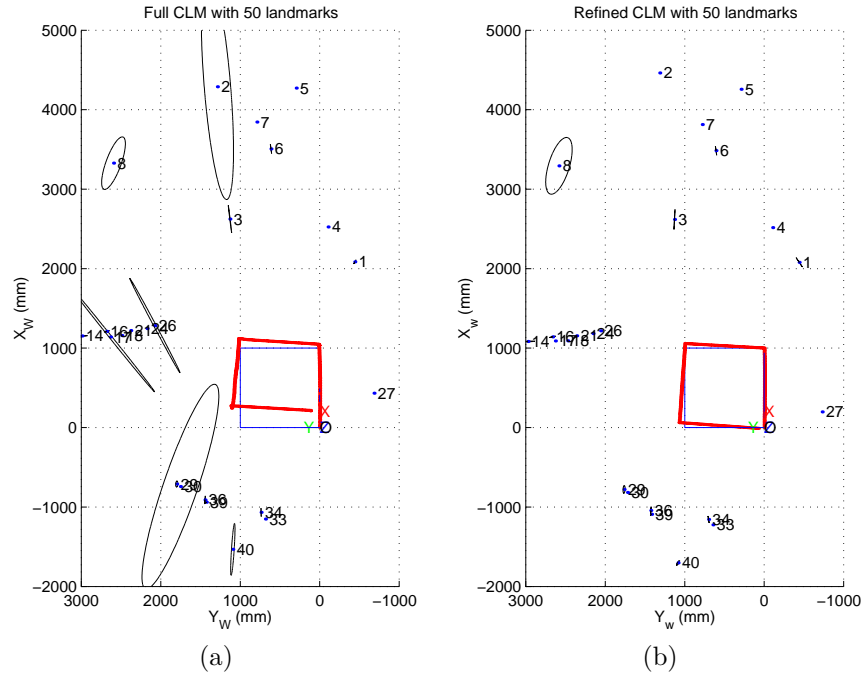


Figure 4: Concurrent Localization and Mapping

assumption on the distribution of the landmark positions, but it does expect white distribution of dead reckoning and sensor errors.

One of the main contributions of this work resides on the formulation of strength measures used to validate landmarks on the basis of their temporal and spatial dispersions. Temporary landmarks and those coming from noisy sensor readings are pruned from the map as their existence state diminishes over time. On the other hand, those groups of landmarks that are repeatedly seen are considered stronger indicators of the structure of the environment.

The combination of landmark strength validation for map updating and Kalman filtering for position estimation provides a suitable platform for the learning of indoor dynamic environments.

Acknowledgments

This work was partially funded by CICYT project DPI 2001-2223, CONACYT and CETYS.

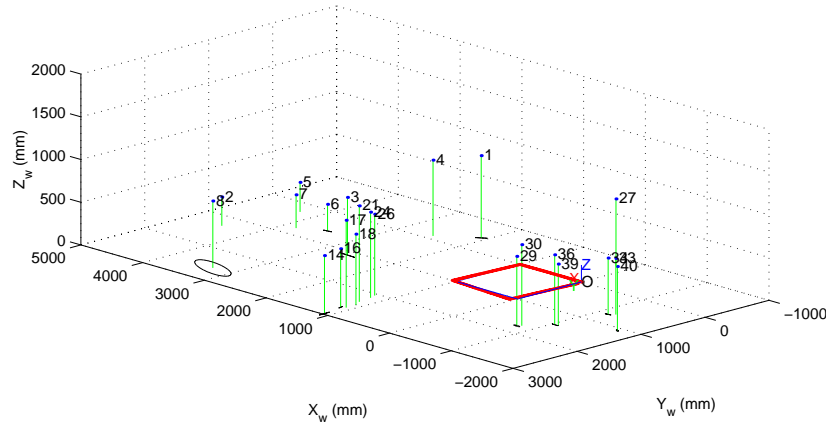


Figure 5: Three-dimensional View

References

1. J. Andrade-Cetto and A. Sanfeliu, "Learning of dynamic environments by a mobile robot from stereo cues," in *Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst.*, Baden-Baden, 2001, pp. 305–310.
2. G. C. Anousaki and K. J. Kyriakopoulos, "Simultaneous localization and map building for mobile robot navigation," *IEEE Robot. Autom. Mag.* **6**, 3 (1999) 42–53.
3. R. Araújo and A. T. de Almeida, "Learning sensor-based navigation of a mobile robot in unknown worlds," *IEEE T. Syst. Man Cyber. B* **29**, 2 (1999) 164–178.
4. A. Arleo, J. del R. Millán and D. Floreano, "Efficient learning of variable-resolution cognitive maps for autonomous indoor navigation," *IEEE T. Robot. Automat.* **15**, 6 (1999) 990–1000.
5. E. Brookner, *Tracking and Kalman Filtering made Easy*, John Wiley & Sons, New York, 1998.
6. J. A. Castellanos, J. M. M. Montiel, J. Neira and J. D. Tardós, "The SPMAP: A probabilistic framework for simultaneous localization and map building," *IEEE T. Robot. Automat.* **15**, 5 (1999) 948–952.
7. H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE T. Robot. Automat.* **17**, 2 (2001) 125–137.
8. R. Deriche and T. Blaszká, "Recovering and characterizing image features using an efficient model based approach," in *Proc. 8th IEEE Conf. Comput. Vision Pattern Recog.*, New York, 1993, pp. 530–535.
9. R. Deriche and G. Giraudon, "A computational approach for corner and vertex detection," *Int. J. Comput. Vision* **10**, 2 (1993) 101–124.

10. M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE T. Robot. Automat.* **17**, 3 (2001) 229–241.
11. T. Duckett and U. Nehmzow, "Mobile robot self-localization and measurement of performance in middle-scale environments," *Robot. Auton. Syst.* **24**, 1-2 (1998) 57–69.
12. D. Fox, W. Burgard and S. Thrun, "Markov localization for mobile robots in dynamic environments," *J. Artif. Intell. Res.* **30** (1999) 391–427.
13. P. Gaussier, S. Leprêtre, M. Quoy, A. Revel, C. Joulian and J. P. Banquet, "Experiments and models about cognitive map learning for motivated navigation," in *Interdisciplinary Approaches to Robot Learning*, eds. J. Demiris and A. Birk, World Scientific, volume 28 of *World Scientific Series in Robotics and Intelligent Systems*, 2000 pp. 53–94.
14. Y. D. Kwon and J. S. Lee, "A stochastic map building method for mobile robot using 2-d laser range finder," *Auton. Robot.* **7**, 2 (1999) 187–200.
15. D. Lee and M. Recce, "Quantitative evaluation of the exploration strategies of a mobile robot," *Int. J. Robot. Res.* **16**, 4 (1997) 413–447.
16. J. J. Leonard, H. F. Durrant-Whyte and I. J. Cox, "Dynamic map building for an autonomous mobile robot," *Int. J. Robot. Res.* **11**, 4 (1992) 286–292.
17. P. S. Maybeck, *Stochastic Models, Estimation, and Control*, volume 1, Academic Press, New York, 1979.
18. A. Ohya, A. Kosaka and A. C. Kak, "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing," *IEEE T. Robot. Automat.* **14**, 6 (1998) 969–978.
19. G. Oriolo, G. Ulivi and M. Vendittelli, "Real-time map building and navigation for autonomous robots in unknown environments," *IEEE T. Syst. Man Cyber. B* **28**, 3 (1998) 316–332.
20. S. Thrun, "Bayesian landmark learning for mobile robot localization," *Mach. Learn.* **33**, 1 (1998) 41–76.
21. G. Welch and G. Bishop, "An introduction to the Kalman filter," Technical Report TR 95-041, Dept. of Comput. Sci., Univ. of North Carolina at Chapel Hill, 1995.