# Learning of Dynamic Environments by a Mobile Robot from Stereo Cues

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,sanfeliu@iri.upc.es

## Abstract

*A system that builds a three-dimensional map of an indoor environment for a mobile robot is presented. The approach uses visual features extracted from stereo images as landmarks. A learning rule associated with each landmark is used to compute its existence state. New landmarks are merged into the map and transient landmarks are removed from the map over time. The location of the landmarks in the map is continuously refined from observations. The position of the robot is estimated by combining sensor readings, motion commands, and the current map state by means of an Extended Kalman Filter. The combination of neural network principles for map updating and Kalman filtering for position estimation allows for robust map learning of indoor dynamic environments.*

**Keywords**: *Map learning, mobile robot navigation, topological maps.*

## 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, at least the descriptors and methods that will be used 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 in cluttered environments [8, 12]. 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 indoors map by a mobile robot:

1. Dead reckoning. As a robot moves, its global position estimate from encoder readings accumulates drift errors, 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.
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 made typically by updating grid maps of obstacles. Recent contributions on grid-based map building include [2, 13]. An extension to these methods includes sensor signatures on each grid cell for later attempts at robot localization [7]. Probabilistic methods for obstacle parameterization and robot localization have also been suggested [5, 10, 14]. A technique used to refine a simulated graph-based map using linear components in a truss is presented in [9]. Some authors have even recently proposed the use of goal oriented cognitive maps to learn the relationship between successively explored places [3, 4]. However, all of these methods are usually limited in that it is not possible to deal with changing environments.

A methodology for the construction and update of a dynamic topological map by a mobile robot is presented. Unlike grid-based techniques, it is scale independent. It was designed so that map updating can occur even in changing environments, exploiting the relationship between neighboring landmarks. It does not make any assumption on the distribution of the landmark positions, but it does expect white distribution of dead reckoning and sensor errors. The system architecture is shown in Fig. 1, and it includes three distinctive modules: sensing, robot localization, and map updating.

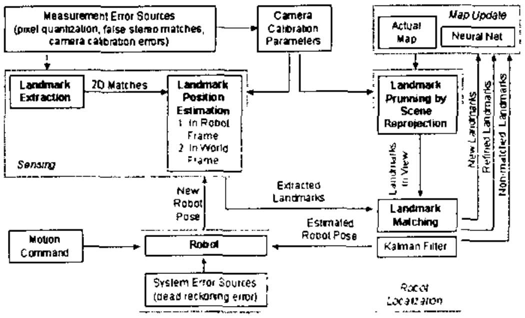

Figure 1. System Architecture

## 2. Landmark Extraction

The extraction of landmarks from the environment relies solely on visual information. A salient feature locator was implemented based on Beaudet's cornerness measure [6], and a refinement using a variance descent approach. These salient features are pairwise matched in each stereo set by correlation and by the enforcement of epipolar constraints. Each feature's 3-D position $\mathbf{z}_i^R$ with respect to the robot is reconstructed from stereo geometry, and together with an associated vector of appearance properties $\zeta_i$, constitute a landmark. The appearance properties, which are used to validate future map landmark matches, include the pixel gray-level mean and distribution over a small window, and the energy of the feature computed from the cornerness measure.

The landmark position estimates and their correlational information will help in building the map, as well as in positioning the robot within this map by minimizing the localization error in the least squares sense. To this aim, we must define the landmark position with respect to the robot in terms of the current estimated robot pose, the absolute landmark position in world coordinates, and the uncertainty in sensor measurement:

$$\mathbf{z}_i^R(k) = h(\mathbf{x}(k), \mathbf{z}_i^W, \mathbf{v}(k)) \tag{1}$$

A noise-free approximate measure of this quantity is given by

$$\begin{aligned}
\tilde{\mathbf{z}}_i^R(k) &= \mathbf{R}^{\mathrm{T}}(k)\mathbf{z}_i^W - \mathbf{R}^{\mathrm{T}}(k)\mathbf{t}(k) && (2)\\
\mathbf{R}(k) &= \begin{bmatrix} \cos(\theta(k)) & -\sin(\theta(k)) & 0 \\ \sin(\theta(k)) & \cos(\theta(k)) & 0 \\ 0 & 0 & 1 \end{bmatrix} && (3)\\
\mathbf{t}(k) &= [x(k), y(k), 0]^{\mathrm{T}} && (4)
\end{aligned}$$

and a linearized version can be expressed as a Taylor Series with the higher order terms dropped, i.e.,

$$\mathbf{z}_i^R(k) \approx \tilde{\mathbf{z}}_i^R(k) + \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\tilde{\mathbf{x}}(k), \mathbf{z}_i^W, 0)(\mathbf{x}(k) - \tilde{\mathbf{x}}(k)) + \mathbf{v}(k) \tag{5}$$

In practice we do not know the true values for the robot pose at time step $k$, nor the real landmark positions. It is only through Eqs. 2 and 5 that we are able to refine an estimate for the robot pose $\tilde{\mathbf{x}}(k)$.

As the reader may suggest by now, we are setting the necessary building blocks for the formulation of an Extended Kalman Filter approach to robot self localization. Before delving into the details of the EKF, we will explain first how the map data structure is preserved.

## 3. Map Update

The updating of the map is a continuous process that takes place during robot navigation. All landmarks $D(k) = \{\mathbf{z}_1^R, \mathbf{z}_2^R, \ldots, \mathbf{z}_l^R\}$ coming from sensor measurements are searched for a match in the previous map state $T(k-1)$. This search is limited to a reduced number of landmarks, i.e., all current map entries are first reprojected into the visual space, and only those map landmarks that fall within the field of view at the current robot position are considered during the map update process. $T'(k) \subseteq T(k-1)$ being the set of landmarks that fall within the field of view.

If a sensed landmark falls within the uncertainty region of a pruned map landmark, and their appearance properties are highly correlated, then we can consider it a scene-to-map landmark match. Once a match is obtained, the distribution parameters for the uncertainty of the landmark position are updated, as well as its vector of appearance properties. Scene to map landmark matches are contained in $M(k) = D(k) \cap T'(k)$, and the set of new landmarks that must be added to the map is $N(k) = D(k) - M(k)$.

The uncertainty region for the location of a landmark in the map is parameterized by a normal distribution with running mean $\bar{\mathbf{z}}_i^W$ and sample covariance $\mathbf{S}_i$; and to check if a scene landmark falls inside the uncertainty region of a map landmark we use the Mahalanobis distance.

The landmark existence state was first introduced in [1] as a measuring device of how persistent a landmark is in order to be considered a strong reference for environment representation and robot localization. Groups of landmarks repeatedly present on a scene are considered strong indicators of the structure of the environment; and temporary landmarks and those coming from noisy sensor readings are pruned from the map as their existence state diminishes over time. The set $F(k)$ contains the landmarks that are removed from the map during the $k$-th iteration.

We have resorted to neural network principles for the formulation of the landmark existence state because of the

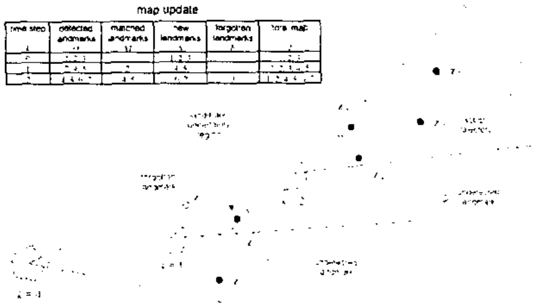

Figure 2. Map Update

```
for each iteration do:
$D(k)$     $\leftarrow$  Landmark extraction.
$T'(k)$    $\leftarrow$  Pruning by map reprojection.
                   Input: $T(k-1), \hat{x}(k-1)$.
$M(k)$     $\leftarrow$  Scene to map landmark matching.
                   Input: $D(k), T'(k), \hat{x}(k-1)$.
$T(k)$     $\leftarrow$  Map augmentation with new landmarks,
                   refinement of matched landmark positions,
                   and pruning of weak landmarks.
                   Input: $N(k), M(k), F(k), T(k-1), \hat{x}(k-1)$.
$\hat{x}(k)$  $\leftarrow$  Robot pose update.
                   Input: $M(k), T(k-1), \hat{x}(k-1)$.
```

Table 1. Algorithm for Map Update

exponential decay properties of the Hebbian learning rule and the possibility to link neighboring landmarks in a networked representation. Within this framework, two approaches have been analyzed for the update of the existence state: first, we consider the landmarks as uncorrelated features that characterize the robot environment; and secondly, their reciprocal relationship is examined[1].

### 3.1. Independent Landmarks

For each landmark in the map, there exists an associated memory cell that will register how persistent, and how old the landmark is. The state of the $i$-th neuron $x_i(k)$ will be considered as the existence state or strength of landmark $i$, and the input to the neuron $\bar{e}_i(k) = \{-1, 1\}$ the landmark identification stamp at time $k$. The proposed update rule equation for the existence of landmark $i$ in the map is

$$x_i(k+1) = \frac{1}{1 + e^{-(\alpha_i \bar{e}_i(k) + \beta_i x_i(k))}} \qquad (6)$$

where $\alpha_i$ is the input weight, used to regulate the contribution of the identification of a landmark in the current scene over the previous map configuration; and $\beta_i$ is the memory weight, used to regulate the contribution of the previous state of a given landmark over its new state.

It should be stressed that the neuron states are only updated for those landmarks in the set $T'(k)$, i.e., the ones that fall within the view area from the current robot position at time step $k$. At this time, we do not evaluate the strength of a landmark in the map if it cannot be projected onto the field of view. Finally, if the existence state $x_i(k)$ falls below a forgetting threshold $t_f$, it means the corresponding landmark has been forgotten, and it is immediately removed from the map.

### 3.2. Correlated Landmarks

The learning rule in Eq. 6 omits the relationships that exist among different landmarks, thus neglecting correlational information. To model these correlations, consider now the following rule for the propagation of activity through the network

$$x_i(k+1) = \frac{1}{1 + e^{-\left(\alpha_i \bar{e}_i(k) + \beta_i x_i(k) + \sum_{j \in T'(k)} w_{ji}(k) x_j(k)\right)}} \qquad (7)$$

and the long-term learning expression $w_{ji}(k+1) = w_{ji}(k) + \gamma x_i(k) x_j(k)$; with $\gamma$ a positive constant that determines the speed of learning. It is common practice to normalize the sum of weights into a state node $i$ to restrict them from growing indefinitely over time.

To this point, the search for a scene to map landmark match takes place one landmark at a time. We will see now how by considering groups of consistently present landmarks (landmark sets) we can speed up this search. The grouping of landmarks into landmark sets comes as a byproduct of our existence state learning algorithm. Say, if landmarks $i$ and $j$ appear consistently in a scene, their states will be active at the same time, and their weights $w_{ij}(k)$ and $w_{ji}(k)$ will see a considerable increase. A large value for a connection from landmark $i$ to landmark $j$ in the network implies strong landmark correlation. Conversely, if their existence states are not active simultaneously, their learned weights will not see much improvement.

To use this correlational information to our advantage during scene to map landmark matching, consider the following strategy. For each map entity, we maintain links to their highly correlated counterparts. These sets of links constitute the aforementioned landmark sets. The search order for a scene to map landmark match is governed by the

[1] In [1] a more complicated model for a neuron was used. However, our experiments have shown that the simpler Hebbian learning rule is easier to fine tune, and still exhibit the desired behavior for the existence state in both the independent and correlated landmark cases.

weights learned on these links. This is, once a scene landmark has been matched to a map landmark, the next scene landmark will be first compared to those in the map known to be correlated to the one previously matched. Provided the scene has not changed considerably, and the robot has not moved a long way on each iteration, this technique allows for fast scene to model landmark matching, thus reducing in average the overall time complexity of the map building algorithm.

Note that different from the rule used in the uncorrelated case, we need now to update not only the existence state, but as much as $|T(k-1)| - |F(k)| + |N(k)|$ weights for each landmark as well. This is an upper bound. In reality, the number of weights for each landmark is close to $|D(k)| - 1$, plus the connections to other landmarks in previous views. The size of the set $D(k)$ can be regulated during feature extraction. By dynamically keeping the sensitivity of the corner detector within a specified range, we can ensure a systematical number of corners extracted from the images at all times.

The proposed map update scheme has the following advantages over other map learning algorithms:

1. The map preserves its topological structure. The prevailing relationships among existing features are their own Euclidean metrics, as well as the learned weights for the correlated case.

2. The map is not limited in resolution, as opposed to grid-based maps. This allows for the modeling of different size environments without the need to modify its general structure.

3. The search for a scene to map landmark match can be sped up if the correlational information among landmarks is considered.

4. The dynamic property of the map allows for the robust modeling of changing or noisy environments. It also refrains the map from growing indefinitely, a situation that could affray with system resources (search speed and memory).

Table 1 summarizes the steps performed during each iteration of the map building process, and Fig. 2 illustrates this process. The figure shows the position of the robot at various time instances. The triangles represent the field of view of the robot, and the landmarks are represented by dots. The thick doted lines connecting some of the landmarks indicate the weight connections on the correlated case for the computation of the landmark existence state.

## 4. Robot Localization

Resorting to the classification presented in [15], probabilistic mobile robot localization techniques can be divided mainly into two groups:

1. Kalman filters that estimate the current robot position from current and previous sensor readings, past position estimates, and motion commands, as well as uncertainty measurements of sensory and motion information, e.g. [7].

2. Markov localization techniques which maintain a position probability density over the whole set of robot poses, e.g. [8].

Kalman filters are typically robust for incremental localization, whereas Markov localization is better suited for global localization. The former technique requires that the initial location of the robot be known, and the latter method usually requires stronger assumptions about the nature of the environment. In this work, we are limited to the construction of an environment map starting from a known position or origin; and since the computation of the actual robot position is very sensitive to the accumulation of dead-reckoning error, we have opted for the use of an Extended Kalman Filter for robot localization.

Continuing the discussion started in Section 2, we will describe now the equations used to update our estimate of the robot pose from landmark measurements and the actual equations that govern the robot motion. Each iteration of the EKF for the update of the robot pose estimate can be divided into a correction step and a prediction step. It is the results from the prediction step that serve as input data to the correction step.

By differentiating Eq. 1 with respect to the robot pose, we obtain the following expression for the Jacobian or measurement innovation matrix for landmark $i$ at time step $k$:

$$\mathbf{H}_i(k) = \left[ \begin{array}{ccc} -\cos(\theta) & -\sin(\theta) & [\dot{\mathbf{R}}^{\top}(\mathbf{z}_i^W - \mathbf{t})]_1 \\ \sin(\theta) & -\cos(\theta) & [\dot{\mathbf{R}}^{\top}(\mathbf{z}_i^W - \mathbf{t})]_2 \\ 0 & 0 & -1 \end{array} \right]_{\substack{\mathbf{x}=\hat{\mathbf{x}}(k) \\ \mathbf{z}_i^W=\hat{\mathbf{z}}_i^W(k)}} \tag{8}$$

where the notation $[\cdot]_j$ represents the $j$-th vector element.

During the correction step, the following computations take place for each landmark in $M(k)$ (recall that $M(k)$ is the set of scene to map landmark matches as specified in Sec. 3):

First, the Kalman Filter Gain is computed with

$$\mathbf{K}_i = \mathbf{P}\mathbf{H}_i^{\top} \left(\mathbf{H}_i\mathbf{P}\mathbf{H}_i^{\top} + \mathbf{R}^{\top}\mathbf{S}_i\mathbf{R}\right)^{-1} \tag{9}$$

where $\mathbf{S}_i$ is the measurement residual error covariance at time $k$ for landmark $i$, and $\mathbf{R}^{\top}\mathbf{S}_i\mathbf{R}$ is the base change that will express this error 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,

$$\delta \mathbf{x}_i(k) = \mathbf{K}_i(k)(\mathbf{z}_i^R(k) - \mathbf{h}(\tilde{\mathbf{x}}(k), \bar{\mathbf{z}}_i^W(k), 0)) \qquad (10)$$

Eq. 10 shows how each of the observed landmarks contributes to the correction of the robot pose estimate $\tilde{\mathbf{x}}_i(k)$. Note that $\bar{\mathbf{z}}_i^W(k)$ is the best approximation we have to the real landmark position in world coordinates at time step $k$, and that the measurement error $\mathbf{z}_i^R(k) - \mathbf{h}(\tilde{\mathbf{x}}(k), \bar{\mathbf{z}}_i^W(k), 0)$ 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 estimate for the landmark position $\bar{\mathbf{z}}_i^W$ is guaranteed to converge to its true value.

In the typical formulation of the EKF, it is always known where the observations come from. This is, all features can be observed at the same time, and can be matched from one step to the next. They form what is known as the observation vector, and relate to the state variable (the robot pose) through just one equation similar to Eq. 1. If we were to consider landmark correlations during an update iteration on the robot pose, we should stack together all measurements $\mathbf{z}_i^R$ in a $3n$ dimensional vector, and obtain a $3n \times 3n$ Jacobian matrix and a similarly large Kalman gain. However, the robot can only see a few of these features at a time, and only after matching their appearance properties by correlation to entries in the map, we would know their position within this large observation vector. Also, the size of $n$ (the total number of landmarks) is not known a priori. For this reason, we are forced to consider the contribution from each observation separately, and compute filter gains and estimate updates for each of them.

To combine the contribution from all matched landmarks for the estimate of the robot pose we have opted for a weighted sum of the individual contributions.

$$\tilde{\mathbf{x}}(k) = \tilde{\mathbf{x}}(k) + \frac{1}{|M|} \sum_M \delta \mathbf{x}_i(k) \qquad (11)$$

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

$$\mathbf{P}(k) = \left( \mathbf{I} - \frac{1}{|M|} \sum_M \mathbf{K}_i(k) \mathbf{H}_i(k) \right) \mathbf{P}(k) \qquad (12)$$

To predict the behavior of the system, we need to add the motion command vector $\mathbf{u}(k)$, and system noise $\mathbf{Q}$ for the next time step. This is,

$$\begin{aligned} \tilde{\mathbf{x}}(k+1) &= \tilde{\mathbf{x}}(k) + \mathbf{u}(k) && (13) \\ \mathbf{P}(k+1) &= \mathbf{P}(k) + \mathbf{Q}(k) && (14) \end{aligned}$$

The robot motion 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 $\mathbf{Q}$.

The evaluation of Eqs. 9-14 after each image processing step completes an iteration of the Kalman filter, used to refine our robot pose estimate $\tilde{\mathbf{x}}(k)$. The reader is referred to [11, 16] for a detailed explanation on Kalman Filtering techniques. Our formulation follows closely that of [16].

## 5. Experiments

A series of controlled experiments were performed to test the viability of the proposed system. A set of 60 image pairs of an easily identifiable stationary landmark were taken by our mobile robot MARCO at a distance of approximately 50cm. Then, the robot was commanded a motion of 10cm backwards, and a rotation of 5° was exerted counterclockwise, both at time step $k = 61$; and a new set of 60 image pairs was acquired. In this experiment we wanted to show two things. Firstly, how the projection of the measured landmark position in the world reference frame can vary significantly after a motion command due to dead reckoning error. And secondly, how the estimate for the landmark position improves during each iteration of the filter.

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

Fig. 3 shows the detected landmark, as well as its uncertainty region from a point of view similar to that of the robot. The × marks indicate the measured landmark positions projected into the world reference frame. The dotted trajectory indicates the adjustment of the mean landmark position; and the ellipse indicates the distribution of the sampled measures around their current mean. Note how the landmark position measurements are separated by an apparently constant value along the $Z_W$ axis. This is due to variations by one row in the localization of the landmarks in the images.

A second experiment was performed on a real scenario. Fig. 4 shows a group of matched salient features from one of the stereo images. The hollow boxes show the features extracted from the image, whereas the filled boxes correspond to features that have been matched properly in the stereo pair. Fig. 4 shows also a top view of the estimated position of the matched landmarks with respect to the robot as well as their position error covariance estimate in the form of uncertainty ellipses. The straight lines indicate the correspondence of scene to map landmarks. The updated robot position is also shown in the figure.

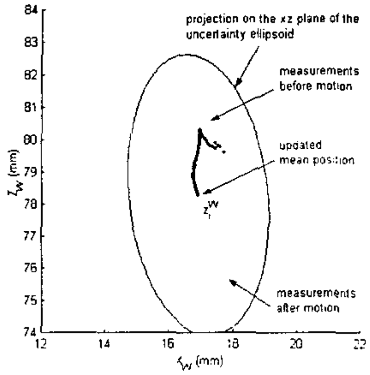

Figure 3. Detected landmark position estimate

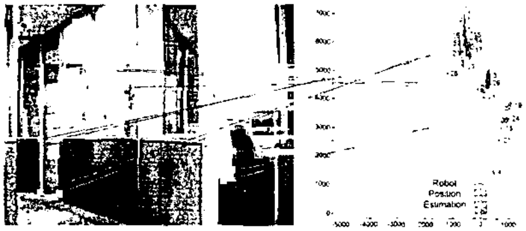

Figure 4. Map building over a real scenario

## 6. Conclusions

The methodology for the construction and update of a dynamic map 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 relationships existing among neighboring landmarks, and the persistence of each landmark in the scene. It does not make any assumption on the distribution of the landmark positions, but it does expect white distribution of dead reckoning and sensor errors.

We propose a novel formulation of the landmark existence state, a measuring device of how persistent a landmark is in order to be considered a strong reference for environment representation and robot localization. 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.

Due to the fact that the robot can only see a limited number of landmarks during each iteration, we have considered the independent contribution from each observation for the updating of the robot position. The combination of neural network principles for map updating and Kalman filtering for position estimation provides a suitable platform for the learning of indoor dynamic environments.

## Acknowledgments

Work partially funded by CICYT project TAP98-0473, CONACYT and CETYS.

## References

[1] J. Andrade-Cetto and A. Sanfeliu. Topological map learning for a mobile robot in indoor environments. In J. S. Sánchez and F. Pla, editors, *Proc. 9th Spanish Sym. Pattern Recog. Image Anal.*, volume 1, pages 221–226, Benicasim, 2001.

[2] G. Anousaki and K. Kyriakopoulos. Simultaneous localization and map building for mobile robot navigation. *IEEE Robot. Autom. Mag.*, 6(3):42–53, 1999.

[3] R. Araújo and A. de Almeida. Learning sensor-based navigation of a mobile robot in unknown worlds. *IEEE T. Syst. Man Cy. B*, 29(2):164–178, 1999.

[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):990–1000, 1999.

[5] J. Castellanos, J. Montiel, J. Neira, and J. Tardós. The spmap: A probabilistic framework for simultaneous localization and map building. *IEEE T. Robot. Automat.*, 15(5):948–952, 1999.

[6] R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *Int. J. Comput. Vision*, 10(2):101–124, 1993.

[7] T. Duckett and U. Nehmzow. Mobile robot self-localization and measurement of performance in middle-scale environments. *Robot. Auton. Syst.*, 24(1-2):57–69, 1998.

[8] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *J. Artif. Intell. Res.*, 30:391–427, 1999.

[9] M. Golfarelli, D. Maio, and S. Rizzi. Correction of dead-reckonning errors in map building for mobile robots. *IEEE T. Robot. Automat.*, 17(1):37–47, 2001.

[10] Y. Kwon and J. Lee. A stochastic map building method for mobile robot using 2-d laser range finder. *Auton. Robot.*, 7(2):187–200, 1999.

[11] P. Maybeck. *Stochastic Models, Estimation, and Control*, volume 1. Academic Press, New York, 1979.

[12] 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):969–978, 1998.

[13] G. Oriolo, G. Ulivi, and M. Vendittelli. Real-time map building and navigation for autonomous robots in unknown environments. *IEEE T. Syst. Man Cy. B*, 28(3):316–332, 1998.

[14] S. Thrun. Bayesian landmark learning for mobile robot localization. *Mach. Learn.*, 33(1):41–76, 1998.

[15] S. Thrun. Probabilistic algorithms in robotics. *AI Mag.*, 21(4):93–109, 2000.

[16] 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.