Robot Environment Modeling via Principal Component Regression

Nikos Vlassis Ben Kröse

RWCP Autonomous Learning Functions SNN
Dept. of Computer Science, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
The Netherlands
E-mail: {vlassis, krose}@wins.uva.nl

Abstract

A key issue in mobile robot applications involves building a map of the environment to be used by the robot for localization and path planning. In this paper we propose a novel framework for robot map building which is based on principal component regression, a statistical method for extracting low-dimensional dependencies between a set of input and target values. A supervised set of robot positions (inputs) and associated high-dimensional sensor measurements (targets) are assumed. A set of globally uncorrelated features of the original sensor measurements are obtained by applying principal component analysis on the target set. A parametrized model of the conditional density function of the sensor features given the robot positions is built based on an unbiased estimation procedure that fits interpolants for both the mean and the variance of each feature independently. The simulation results show that the average Bayesian localization error is an increasing function of the principal component index.

Keywords: *robot environment modeling, principal component analysis, generalized linear regression, Bayesian inference, unbiased variance prediction, robot localization.*

1 Introduction

Useful mobile robots must be able to navigate from their current state toward a desired state. Most often the state of a mobile robot is expressed as a position in the configuration space. However, relying only on the odometry, the robot will not end up accurately at its desired position. Sensors looking at the external world will have to be used for position estimation. Because sensor signals are noisy, a wide variety of probabilistic methods have been developed to obtain a robust estimate of the position of the robot given its sensory inputs, often based on a (learned) relation between position and sensor pattern [2, 9, 13, 12].

Instead of describing the state in the configuration space it is also possible to describe the state in the sensor space: the space spanned by the sensor values or features derived from these [6]. In this case the localization is more straightforward, but the desired state must also be given in the sensor domain.

In both approaches a model has to be made of the sensor data. This introduces a problem. Current sensor technology provides high-dimensional data vectors such as images or range profiles. For an accurate modeling of these data we should need an extremely large set of learning samples. It is therefore essential that the dimensionality of the data is reduced adequately before the modeling step. Feature extraction thereby becomes more and more important in relation with increasing sensor capabilities. An important question now is how to determine the best set of features.

A very general approach to feature extraction is the (non)linear projection of high-dimensional data onto low dimensional subspaces. In a number of robotic applications linear projections (principal component analysis) have been proposed to preprocess image data [8, 5] or range data [3]. In landmark based navigation a set of landmarks (usually predefined by the designer) can be regarded as nonlinear features. The best set of features should generally minimize the localization error.

In this paper we first introduce a parametrized model of the conditional density function of a set of linear sensor features and show how this model can be used for localization. We then investigate how the localization error depends on the selection of linear features in the model.

2 Feature extraction with PCA

Principal component analysis (PCA) [4] is a statistical method for reducing the dimensionality of a data set of correlated variables while retaining most of the original variation of the data. In robotics the method has been used for extracting linear features from images [8] and range sensor data [3].

Assume a set of N d-dimensional range sensor data $\{\mathbf{z}_n\}$, forming the rows of an $N \times d$ matrix $\mathbf{Z}$. The measurements are collected at respective robot positions $\{\mathbf{x}_n\}$, while each dimension d of the data corresponds to the measured distance between the robot and a nearby obstacle along the respective direction.[1] To facilitate the computations we first normalize the column means to zero by subtracting from each row the mean of the sensor measurements. PCA projects the data onto a lower ($q < d$)-dimensional space in such a way that the new q variables, called principal components or features, are globally uncorrelated, while most of the variation of the original data set is preserved.

The numerically most accurate way to do this is by taking the singular value decomposition [10] $\mathbf{Z} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$ of the sensor matrix $\mathbf{Z}$, where $\mathbf{V}$ is a $d \times q$ orthogonal matrix with columns corresponding to the q eigenvectors with largest eigenvalues of the covariance matrix of $\mathbf{Z}$. Then the rows of the $N \times q$ matrix $\mathbf{Y} = \mathbf{Z}\mathbf{V}$ are the projections of the original d-dimensional points to the new q-dimensional feature space, and this projection is optimal under the trace and also the determinant of the projected covariance matrix.

3 Principal component regression

Principal component regression [4, ch. 8] is an extension to the basic PCA framework that also involves regression. One of the variants of the method involves modeling the relationships between a set of independent variables and an other set of projected variables obtained with PCA from an original data set. In our case, the independent or input variable is considered the robot position vector $\mathbf{x}$, while the target variables are the projected features y^i, with $i = 1, \ldots, q$. Our task is to define appropriate regression surfaces that realize a mapping from the robot position to the projected feature space, and then solve those regression problems based on the supervised data $\{\mathbf{x}_n, y_n^i\}$, for $i = 1, \ldots, q$ and $n = 1, \ldots, N$ (note that each robot position corresponds to one sensor profile, and this profile is projected by PCA to its q features).

Assuming the dimension of the robot workspace is 2, we fit the interpolating 2-D surfaces to the feature space by ordinary least squares [10]. This implies fitting each feature separately. For a feature y we assume a generalized linear mapping f formed by a set of K fixed two-dimensional Gaussian basis functions $\phi_k(\mathbf{x})$ with centers regularly distributed over the workspace and spreads half the distance between neighboring centers so that the functions overlap

$$f(\mathbf{x}) = \sum_{k=1}^{K} a_k \phi_k(\mathbf{x}) = \mathbf{a}^T \boldsymbol{\phi}(\mathbf{x}), \qquad (1)$$

where $\mathbf{a} = [a_1, \ldots, a_K]^T$ are the parameters of the mapping.

The above generalized linear mapping allows for easy computation of the parameter vector $\mathbf{a}$ for each feature as we show in the next section. Also, it does not impose severe overhead because the dimension of the robot workspace is small. Alternatively, nonlinear mappings based on sigmoid basis functions (feedforward neural networks) can be used and trained with nonlinear optimization [11].

4 Modeling the full conditional density

For each feature the above mapping f induces a parametrized form of the average $E[y|\mathbf{x}]$ of a feature y given a particular position of the robot $\mathbf{x}$. However, a much richer source of information is provided by modeling the full conditional density $p(y|\mathbf{x})$ of the feature given the robot position $\mathbf{x}$.

We model the density $p(y|\mathbf{x})$ of each feature as a univariate Gaussian centered on the parametrized mean $f(\mathbf{x})$ and having variance $s(\mathbf{x})$ also depending on $\mathbf{x}$. Allowing the variance to be a function of the inputs $\mathbf{x}$ allows the mapping to be more realistic than assuming equal variances over the workspace, while it also provides a compact, parametrized representation. We can write the conditional density for a feature y as

$$p(y|\mathbf{x}) = \frac{1}{s(\mathbf{x})\sqrt{2\pi}} \exp\left\{\frac{-[y - f(\mathbf{x})]^2}{2s^2(\mathbf{x})}\right\}, \qquad (2)$$

where for the variance we choose a parametrization similar to the means by using a set of M fixed basis

[1] For visualization purposes we assume throughout that the dimension of the robot workspace is 2, i.e., we ignore the orientation of the robot. It's straightforward to extend our method to 3-dimensional workspaces.

functions $\psi_m(\mathbf{x})$

$$s^2(\mathbf{x}) = \exp\left[\sum_{m=1}^{M} b_m \psi_m(\mathbf{x})\right] = \exp[\mathbf{b}^T \boldsymbol{\psi}(\mathbf{x})]. \quad (3)$$

The exponent ensures that the variance is always positive.

From such a parametrized model for the densities of the individual features we can approximate the joint conditional density $p(\mathbf{y}|\mathbf{x})$ of the feature vector $\mathbf{y}$ given a robot position $\mathbf{x}$ as the product of the marginal densities

$$p(\mathbf{y}|\mathbf{x}) = \prod_{\forall y} p(y|\mathbf{x}), \quad (4)$$

where the product involves all features, i.e., all q coordinates of $\mathbf{y}$. Although this approximation has been adopted also by other researchers [2, 9, 12], it may lead to loss of information since it ignores potential local (i.e., input-dependent) correlations between features. However, in the present context the approximation is more realistic since PCA ensures global uncorrelatedness between features.[2] Having this density allows us to globally localize the robot in real time by using some global localization procedure, e.g., Markov localization [2].

5 Estimating the parameters

We describe here a method for estimating the parameters $\mathbf{a}$ and $\mathbf{b}$ in (1) and (3) needed to parametrize the conditional density (2) of a feature y.

The likelihood of the training set D of robot positions $\mathbf{x}_n$ and respective values $y_n, n = 1, \ldots, N$, of a feature y, written as a function of the unknown parameter vectors $\mathbf{a}$ and $\mathbf{b}$ reads[3]

$$p(D|\mathbf{a},\mathbf{b}) = \prod_{n=1}^{N} p(y_n|\mathbf{x}_n) = \frac{1}{(2\pi)^{N/2}\prod_n s(\mathbf{x}_n)} \exp\left\{-\frac{1}{2}\sum_n \frac{[y_n - f(\mathbf{x}_n)]^2}{s^2(\mathbf{x}_n)}\right\}, \quad (5)$$

with $f(\mathbf{x})$ and $s(\mathbf{x})$ from (1) and (3), respectively.

In order to obtain an unbiased estimate of $\mathbf{a}$ and $\mathbf{b}$ we must apply an iterative two-level procedure [7].

[2]Still the approximation can be considered crude. We have recently implemented in our group the more general case of an input-dependent covariance matrix that captures local correlations between features.

[3]Note that the density $p(\mathbf{x}_n)$ is independent of the parameters and thus dropped from the definition of the likelihood.

At the first level of inference we assume known variance parameters $\mathbf{b}$ and maximize (5) with respect to $\mathbf{a}$. Taking logarithms this is equivalent to minimizing with respect to $\mathbf{a}$ the χ^2 quantity

$$\chi^2(\mathbf{a},\mathbf{b}) = \sum_n \frac{[y_n - f(\mathbf{x}_n)]^2}{s^2(\mathbf{x}_n)} = \sum_n \frac{\left[y_n - \sum_k a_k \phi_k(\mathbf{x}_n)\right]^2}{s^2(\mathbf{x}_n)}, \quad (6)$$

corresponding to a chi-square fitting problem. The optimal parameter vector $\hat{\mathbf{a}}$ is then [10]

$$\hat{\mathbf{a}} = (\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T\mathbf{y}, \quad (7)$$

where $\mathbf{y}$ is the $N \times 1$ vector with elements $\mathbf{y}_n = y_n/s(\mathbf{x}_n)$ and $\mathbf{F}$ is the $N \times K$ design matrix with elements $\mathbf{F}_{n,k} = \phi_k(\mathbf{x}_n)/s(\mathbf{x}_n)$.

At the second level of inference we seek estimates for the parameter vector $\mathbf{b}$ of the input-dependent noise variance $s^2(\mathbf{x})$. To get an unbiased estimate of $\mathbf{b}$ we have to maximize with respect to $\mathbf{b}$ the marginal likelihood obtained by integrating the means out of the joint likelihood (5) as

$$p(D|\mathbf{b}) = \int p(D|\mathbf{a},\mathbf{b})\, p(\mathbf{a})\, d\mathbf{a}, \quad (8)$$

where $p(\mathbf{a})$ is a problem-dependent prior set on the parameters $\mathbf{a}$, introducing some prior knowledge about the distribution of the $\mathbf{a}$ values. For simplicity, we assume here a Gaussian prior with a very large variance, showing no preference for any particular values of $\mathbf{a}$. To facilitate the above integration we expand the χ^2 function (6) in Taylor series around the maximum likelihood estimate $\hat{\mathbf{a}}$ from the first level of inference

$$\chi^2(\mathbf{a},\mathbf{b}) = \chi^2(\hat{\mathbf{a}},\mathbf{b}) + \frac{1}{2}(\mathbf{a}-\hat{\mathbf{a}})^T\mathbf{A}(\mathbf{a}-\hat{\mathbf{a}}), \quad (9)$$

with $\mathbf{A}$ being the Hessian matrix of the χ^2 function with respect to $\mathbf{a}$. Note that the first-order derivative term in the Taylor expansion vanishes since $\hat{\mathbf{a}}$ by construction minimizes $\chi^2(\mathbf{a},\mathbf{b})$. To compute $\mathbf{A}$ we differentiate (6) twice with respect to parameters a_k and a_l to get

$$\mathbf{A} = 2\mathbf{F}^T\mathbf{F} \quad (10)$$

which is also a function of $s(\mathbf{x}_n)$ and thus of $\mathbf{b}$. Assuming in (8) a Gaussian prior $p(\mathbf{a})$ with very large variance and using (9), (6), and (5) the integral (8) becomes Gaussian on the $\mathbf{a}$ parameters yielding (ignoring constants)

$$p(D|\mathbf{b}) \propto \frac{|\mathbf{F}^T\mathbf{F}|^{-1/2}}{\prod_n s(\mathbf{x}_n)} \exp\left\{-\frac{1}{2}\sum_n \frac{[y_n - \hat{\mathbf{a}}^T\boldsymbol{\phi}(\mathbf{x}_n)]^2}{s^2(\mathbf{x}_n)}\right\}. \quad (11)$$

Taking logarithms and maximizing the formula with respect to the variances after some algebra we get

$$s^2(\mathbf{x}_n) = [y_n - \hat{\mathbf{a}}^T \boldsymbol{\phi}(\mathbf{x}_n)]^2 + \boldsymbol{\phi}^T(\mathbf{x}_n)(\mathbf{F}^T\mathbf{F})^{-1}\boldsymbol{\phi}(\mathbf{x}_n). \quad (12)$$

Finally we estimate $\hat{\mathbf{b}}$ from the new $s^2(\mathbf{x}_n)$ by solving a second least squares problem to get

$$\hat{\mathbf{b}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{s} \quad (13)$$

where $\mathbf{s} = [\log s^2(\mathbf{x}_1), \ldots, \log s^2(\mathbf{x}_N)]^T$ and

$$\mathbf{H}(n, m) = \psi_m(\mathbf{x}_n) \quad (14)$$

the $N \times M$ design matrix of the new least squares problem.

6 Robot localization and PCA

The simplest way to regard the problem of robot localization from sensor data is through a Bayesian perspective: the robot's belief about its position at any moment is encoded in a prior probability quantity $p(\mathbf{x})$. When the robot senses a new feature y it computes first the likelihood of the measurement $p(y|\mathbf{x})$ from the parametrized form of the conditional density (2) and then updates its belief according to Bayes' rule

$$p(\mathbf{x}|y) = \frac{p(y|\mathbf{x})\, p(\mathbf{x})}{\int p(y|\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x}}, \quad (15)$$

where the integral ensures that the new probability quantity integrates to 1. Assuming locally uncorrelated features, the Gaussian assumption for $p(y|\mathbf{x})$ allows us to repeat this procedure for all features in $\mathbf{y}$, leading to updated estimates for the robot's belief.

A measure of 'goodness' of a model for robot localization has been recently proposed in [12] as the average Bayesian localization error when using the model. Assuming the robot is at $\mathbf{x}^*$ observing y^*, the localization error $e(\mathbf{x}^*, y^*)$ after applying the Bayes' rule (15) is computed as

$$\begin{aligned} e(\mathbf{x}^*, y^*) &= \int_{\mathbf{x}} L(\mathbf{x}, \mathbf{x}^*) p(\mathbf{x}|y^*) d\mathbf{x} \\ &= \int_{\mathbf{x}} L(\mathbf{x}, \mathbf{x}^*) \frac{p(y^*|\mathbf{x}) p(\mathbf{x})}{p(y^*)} d\mathbf{x} \end{aligned} \quad (16)$$

where $L(\mathbf{x}, \mathbf{x}^*)$ a loss function between the true $\mathbf{x}^*$ and the estimated $\mathbf{x}$ position, e.g., a linear loss $||\mathbf{x} - \mathbf{x}^*||$. To get an average localization error R we must average the above quantity over all possible $\mathbf{x}^*$ and y^*. The simplest way to do this is by using the empirical distribution of the training set giving rise to a Bayesian expected loss [1]

$$R = E_{\mathbf{x}^*, y^*}[e(\mathbf{x}^*, y^*)] \approx \frac{1}{N} \sum_{n=1}^{N} e(\mathbf{x}_n, y_n). \quad (17)$$

Substituting from (16) and approximating $p(\mathbf{x})$ from the empirical distribution of the $\{\mathbf{x}_n\}$ we get

$$R = \frac{1}{N} \sum_{n=1}^{N} \frac{\sum_{m=1}^{N} L(\mathbf{x}_m, \mathbf{x}_n) p(y_n|\mathbf{x}_m)}{\sum_{m=1}^{N} p(y_n|\mathbf{x}_m)}, \quad (18)$$

with complexity $O(N^2)$.

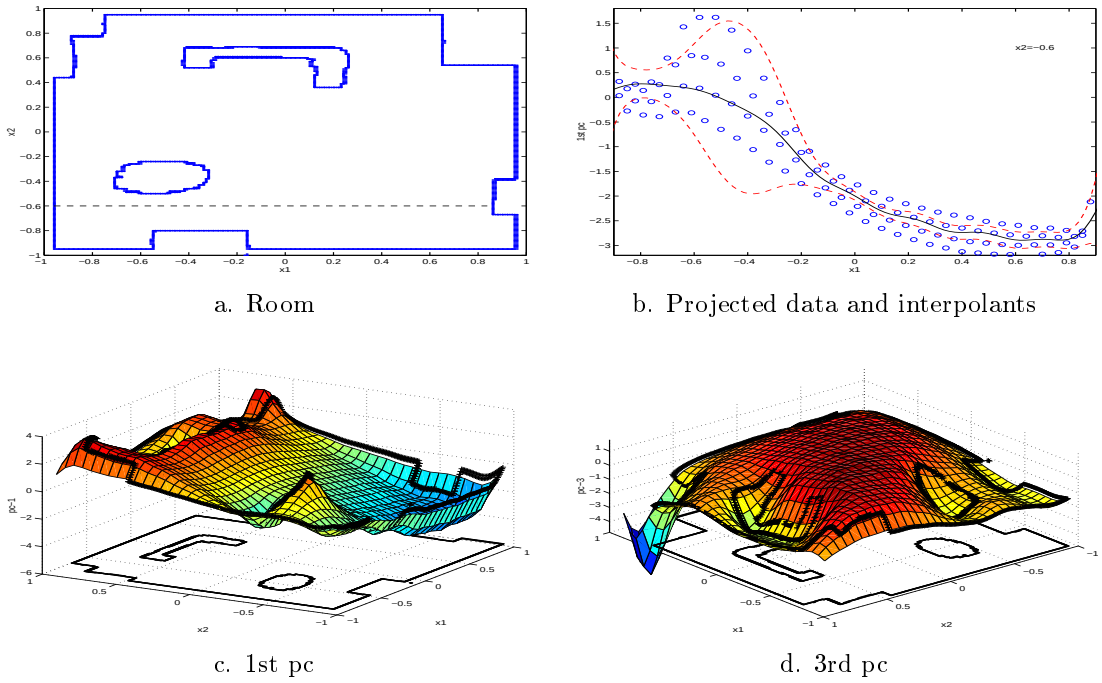
The smaller this quantity is, the better we expect the robot to localize in real time, thus a model for $p(y|\mathbf{x})$ that gives small R should be considered more appropriate for robot localization than one with large R.

Having defined the Bayesian localization error above in terms of the marginal conditional density $p(y|\mathbf{x})$ of each feature y, it is now interesting to investigate how a linear feature projection method like PCA affects this quantity. The parameters of the conditional density of each feature have already been estimated by the method of Section 5, so it is straightforward to compute the localization error for each individual feature by simply introducing the training data into (18). In the next section we show this dependence for a typical indoor configuration.

7 Experiments

A series of experiments were carried out using simulated range data from a robot system which was put randomly in a room with obstacles shown in Fig. 1a. Every datapoint consisted of 360 range measurements over a field of 360 degrees. We obtained datapoints by randomly positioning the robot at about 1000 locations in the room, normalized for the experiments in the range $[-1, 1]$ (the orientation was kept fixed). Note that for many real situations it will be difficult to collect so many supervised learning points, but they can be generated from a smaller set of real measurements as in [3].

We first projected linearly all datapoints onto their eigenspace with the method described in Section 2. With the techniques described in Section 5 we modeled the full conditional density of these projected points given the position of the robot. As an example we show in figure Fig. 1b the values of the projected



a. Room

b. Projected data and interpolants

c. 1st pc

d. 3rd pc

Figure 1: (a) The room set-up. (b) The projected data on the 1st principal component when the robot moves along the line in (a) together with the estimated mean and variance. (c,d) The interpolating surfaces for the 1st and 3rd principal components.

datapoints on the first principal component when the robot moves on the line shown in Fig. 1a (horizontal translation with vertical offset -0.6), together with the obtained parametrization after applying our unbiased estimation procedure in Section 5.

We observe that moving along obstacles, which gives discontinuities in the original sensor measurements, results in high noise estimates in the projected space. With our model it is possible to parametrize the noise variance as a function of inputs (dashed line) around the mean (solid line). For visualization purposes a set of measurements obtained nearby the translation line are also shown in the figure. Note how the noise variance estimate is affected when the robot passes near the oval obstacle on the left.

In Fig. 1c and 1d we show the interpolating surfaces of the parametrized means for the first and the third principal component. We can see that the first principal component varies more with changes in position than the third principal component. This suggests that the first component is a better feature for localization than the third one.

To get a more formal justification of this, we computed the interpolating surfaces for all 360 principal components and then estimated from (18) the localization error that each of them introduces. As shown

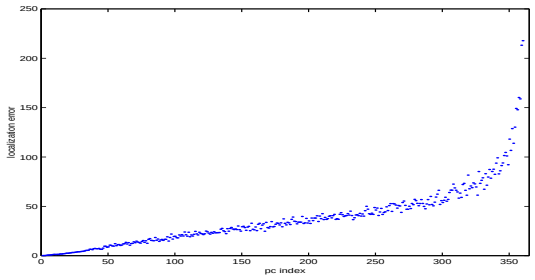


Figure 2: The Bayesian localization error as a function of the principal component index.

in Fig. 2, the localization error is, in general, an increasing function of the principal component index, a property which says that PCA yields linear features with decreasing localization performance.

8 Conclusions-related work

We presented a method for mobile robot environment modeling that is based on principal component regression, a statistical method for building low-dimensional dependencies of input data and targets when the prediction of the projected data is of

interest. We proposed a model for the conditional density of projected features given robot positions, the parametrization of which allows the noise of the data to be spatially varying, a realistic model for mobile robot applications. We described a method that ensures unbiased estimation of the input-dependent noise variance.

An experimental result was also obtained by investigating the relationship between the linear feature extraction method of principal component analysis and the Bayesian localization error described in Section 6. It turned out that the localization error is an increasing function of the principal component index.

A parametrization of the conditional density $p(y|\mathbf{x})$ similar to ours was proposed in [9]. There, the conditional density was defined as a mixture of local Gaussian densities, whose means were linear functions of the robot positions and the variances constant per component. The assumption of linearity, although weaker than our nonlinear parametrization, was justified by the use of a small set of raw proximity sensor measurements which, for planar rooms, exhibit locally linear behavior.

In [3] the method of PCA was used for extracting a small number of linear features from range sensor profiles, while instead of a continuous nonlinear mapping the authors proposed a discretization of both the workspace and the sensor space and then using a tabulated function for modeling their dependencies.

In [2, 13] a separate model for the conditional density was assumed for each cell of the discretized workspace, i.e., a single Gaussian and a mixture of Gaussians, respectively. Finally, in [12] a local conditional density was assumed that was computed by a nearest neighbors approach.

References

[1] J. O. Berger. *Statistical decision theory and Bayesian analysis*. Springer-Verlag, New York, 2nd edition, 1985.

[2] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. 13th Nat. Conf. on Artificial Intelligence*, pages 896–901, Portland, Oregon, 1996.

[3] J. L. Crowley, F. Wallner, and B. Schiele. Position estimation using principal components of range data. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, May 1998.

[4] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[5] B. Kröse and R. Bunschoten. Probabilistic localization by appearance models and active vision. In *Proc. ICRA'99, IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, May 1999.

[6] B. J. A. Kröse and M. Eecen. A self-organizing representation of sensor space for mobile robot navigation. In *Proc. IROS'94, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1994.

[7] D. J. C. MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1991.

[8] S. Maeda, Y. Kuno, and Y. Shirai. Active navigation vision based on eigenspace analysis. In *Proc. IROS'97, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1018–1023, 1997.

[9] S. Oore, G. E. Hinton, and G. Dudek. A mobile robot that learns its place. *Neural Computation*, 9:683–699, 1997.

[10] W. H. Press, S. A. Teukolsky, B. P. Flannery, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 2nd edition, 1992.

[11] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, U.K., 1996.

[12] S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1), 1998.

[13] N. Vlassis, G. Papakonstantinou, and P. Tsanakas. Dynamic sensory probabilistic maps for mobile robot localization. In *Proc. IROS'98, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Victoria, B.C., Canada, Oct. 1998.