Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Supervised Linear Feature Extraction for Mobile Robot Localization

Nikos Vlassis* Yoichi Motomura‡ Ben Kröse*

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

‡Electrotechnical Laboratory
Umezono 1-1-4
Tsukuba Ibaraki 305-8568
Japan
e-mail: motomura@etl.go.jp

Abstract

We are seeking linear projections of supervised high-dimensional robot observations and an appropriate environment model that optimize the robot localization task. We show that an appropriate risk function to minimize is the conditional entropy of the robot positions given the projected observations. We propose a method of iterative optimization through a probabilistic model based on kernel smoothing. To obtain good starting optimization solutions we use canonical correlation analysis. We apply our method on a real experiment involving a mobile robot equipped with an omnidirectional camera in an office setup.

1 Introduction

Current trend in mobile robot technology is towards building fully autonomous mobile robots, i.e., robots that can operate without external guidance in unstructured or natural environments. To localize themselves accurately and then plan paths in their workspace the robots must use their perception mechanism, e.g., vision, often in combination with a dead-reckoning device, e.g., an odometer.

From a statistical viewpoint the robot localization task can be regarded as a *prediction* problem. Given an a priori model of the environment and a new sensor observation the task is to predict the position of the robot as accurately as possibly. Such a model, called *map*, is often built through supervised learning from a set of known robot positions–sensor observations [7, 11, 5, 12].

Sensor technology provides high-dimensional data such as images or range profiles. To deal with the abundance and the inherent redundancy in the data (e.g., too many correlated measurements) an appropriate *feature extraction* scheme should precede the modeling step. The extracted features can be natural landmarks, i.e., distinctive features of the environment [1] or landmarks formed by some mathematical transformation on the original observations [11, 5, 12].

In this paper we deal with the latter case, specifically the extraction of *linear* features from omnidirectional image data to be used for map building and localization. Previous work in our group has investigated the use of *principal component analysis* (PCA) for this purpose [5, 12]. However, PCA is an unsupervised method which optimizes a reconstruction error and may not be necessarily good for localization.

In this paper we look for *supervised* linear projections of the robot observations and an appropriate environment model so that the localization performance of the robot is optimized. In the following we describe the proposed model, the localization criterion to optimize, and the optimization method we use to get the optimal features. We demonstrate our method on a real robot equipped with an omnidirectional camera in an office environment. The results show that our method outperforms PCA as a linear feature extraction method for robot localization.

2 The robot localization problem

Imagine a (point) robot at an unknown position $\mathbf{x}^* \in \mathbb{R}^2$ of its two-dimensional workspace, observing a *d*-

0-7803-5886-4/00/$10.00© 2000 IEEE

dimensional vector[1] $\mathbf{z}^*$, e.g., an image. The robot localization problem concerns the prediction of $\mathbf{x}^*$ given $\mathbf{z}^*$.

To deal with the noise inherent in robot sensing we adopt a probabilistic framework. Let $\mathbf{x}$ be a stochastic vector describing the position of the robot and $p(\mathbf{x}|\mathbf{z}^*)$ the conditional density of $\mathbf{x}$ given a robot observation $\mathbf{z}^*$, call it *posterior density* henceforth. The localization problem can be formulated as finding an estimate of the posterior density as peaked as possible to the real robot position $\mathbf{x}^*$.

First we note that due to perceptual alias the posterior density may exhibit multiple modes in different regions of the workspace. In other words, for the same observation $\mathbf{z}^*$ two or more positions in the workspace can be candidates for $\mathbf{x}^*$. This implies that a solution to the localization problem would be to directly model the posterior density $p(\mathbf{x}|\mathbf{z}^*)$ as a mixture of conditional densities and fit it from the data [9].

However, for realistic robot localization this approach is not adequate since we need to integrate old position estimates and actions into a single position estimate. One way to achieve this is by means of the Bayes' rule. We write the posterior density as

$$p(\mathbf{x}|\mathbf{z}^*) = \frac{p(\mathbf{z}^*|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}^*)} \qquad (1)$$

where $p(\mathbf{z}^*|\mathbf{x})$ is the likelihood of the observation given $\mathbf{x}$, $p(\mathbf{x})$ a *prior* density on the robot positions, and $p(\mathbf{z}^*)$ the unconditional density of observations alone. By repeatedly applying (1) using the posterior density as prior for the next estimate, and independently updating the prior from robot actions, we get a localization procedure that can be used for robot navigation. Kalman filters and hidden Markov models are examples of iterative localization procedures. See [11] for details and references.

3 Feature extraction

We see from the above that robot localization requires a model for $p(\mathbf{z}|\mathbf{x})$, the conditional density of observations given robot positions. To build such a model we assume we are given a supervised training set in the form $D = \{\mathbf{x}_n, \mathbf{z}_n\}, n = 1, \ldots, N$, consisting of N pairs of known robot positions with their associated observations.

In pattern recognition there are many ways to fit conditional densities like $p(\mathbf{z}|\mathbf{x})$ from a training set by using parametric, e.g., a neural network, or nonparametric methods [9]. However, in both cases the efficiency of the learning machine is highly influenced by the dimensionality of the observations, and it turns out that for accurate modeling of high-dimensional data we need a very large number of training points [3, ch. 6.12].

This fact suggests reducing the dimensionality of the observations prior to modeling, in other words, extracting appropriate features from the original high-dimensional data to be used for robot localization. We restrict our attention to linear features, thus for a q-dimensional ($q < d$) vector $\mathbf{y}$ extracted from an observation $\mathbf{z}$ we can write

$$\mathbf{y} = \mathbf{W}^T\mathbf{z}, \qquad (2)$$

with $\mathbf{W}$ a $d \times q$ projection (or feature) matrix.

A usual statistical requirement in such problems is that the components of $\mathbf{y}$ must be uncorrelated. This can be interpreted as a geometrical constraint on the matrix $\mathbf{W}$ if the data $\mathbf{z}$ are already *sphered*, i.e., rotated and scaled so that their covariance matrix is the identity matrix. This is always possible and is equivalent to applying principal component analysis and scaling all components to unit variances [6, 4]. The constraint of uncorrelatedness becomes then a constraint of orthonormality[2] for the matrix $\mathbf{W}$

$$\mathbf{W}^T\mathbf{W} = \mathbf{I}_q \qquad (3)$$

where $\mathbf{I}_q$ stands for the q-dimensional identity matrix. In the following we will assume that the original $\mathbf{z}$ points have already been sphered and have zero mean. The mapping (2) under the constraint (3) in effect rotates the sphered $\mathbf{z}$ space and then retains only the first q most 'useful' coordinates for localization.

Since the mapping (2) is deterministic, the robot localization problem can be reformulated in the projected space by building a model of $p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})$ in this space parametrized[3] on $\boldsymbol{\theta}$. Since each observation $\mathbf{z}$ corresponds to some projected vector $\mathbf{y}$, the Bayes rule (1) can be equally applied substituting $\mathbf{y}$ for $\mathbf{z}$, while for building a model of $p(\mathbf{y}|\mathbf{x})$ we can use the transformed training set $D' = \{\mathbf{x}_n, \mathbf{y}_n\}, n = 1, \ldots, N$.

The rest of the paper concerns building such a model and estimating in a supervised manner, i.e., using both the $\mathbf{z}_n$ and $\mathbf{x}_n$ points of the training set, the projection matrix $\mathbf{W}$ that gives good localization. For that we need a 'goodness' measure.

[1] All vectors are assumed column vectors.

[2] Since $E[\mathbf{y}\mathbf{y}^T] = E[\mathbf{W}^T\mathbf{z}\mathbf{z}^T\mathbf{W}] = \mathbf{W}^T E[\mathbf{z}\mathbf{z}^T]\mathbf{W} = \mathbf{W}^T\mathbf{W}$.

[3] In the sequel, the dependence of the model on the parameters will be assumed and thus skipped.

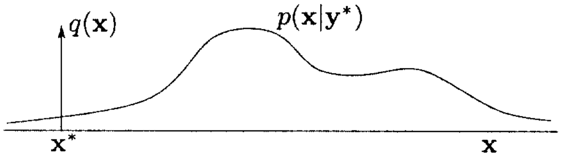

Figure 1: The predicted posterior $p(\mathbf{x}|\mathbf{y}^*)$ and the delta-peaked density $q(\mathbf{x})$ on the real robot position $\mathbf{x}^*$.

4 The localization criterion

In our analysis we consider the worst case where the robot localizes with a flat prior $p(\mathbf{x})$. Moreover, to simplify matters, we assume that the points $\mathbf{x}_n$ in the training set were sampled from the same prior $p(\mathbf{x})$, an assumption which leads to mathematical tractability in the models below.

Suppose the robot is at the position $\mathbf{x}^*$ and observes a vector $\mathbf{y}^*$ derived from (2). Assume also a selected model $p(\mathbf{y}|\mathbf{x})$ which, through Bayes, gives a posterior density $p(\mathbf{x}|\mathbf{y}^*)$. A measure of closeness of the predicted posterior and the real robot position can be derived by taking the *cross-entropy* between a delta-peaked density $q(\mathbf{x})$ on $\mathbf{x}^*$ and the posterior (Fig. 1)

$$c = -\int q(\mathbf{x}) \log p(\mathbf{x}|\mathbf{y}^*) d\mathbf{x} \approx -\log p(\mathbf{x}^*|\mathbf{y}^*), \quad (4)$$

the approximation justified by the fact that $q(\mathbf{x})$ is peaked on $\mathbf{x}^*$.

Averaging the above *loss* function over the joint $\mathbf{x}$–$\mathbf{y}$ space we arrive at the definition of the *conditional entropy* [8] of the positions $\mathbf{x}$ given the projected vectors $\mathbf{y}$

$$H(\mathbf{x}|\mathbf{y}) = -\int\int p(\mathbf{x},\mathbf{y}) \log p(\mathbf{x}|\mathbf{y}) d\mathbf{x} d\mathbf{y} \quad (5)$$

as the expected *risk* using the model $p(\mathbf{y}|\mathbf{x})$ and the mapping (2). The integral can be approximated using the empirical distribution of the training points to get the *empirical* risk

$$R = -\frac{1}{N}\sum_{n=1}^{N} \log p(\mathbf{x}_n|\mathbf{y}_n) = -\frac{1}{N}\sum_{n=1}^{N} \log \frac{p(\mathbf{x}_n,\mathbf{y}_n)}{p(\mathbf{y}_n)} \quad (6)$$

which can also be regarded as the negative log-likelihood of the training set with respect to the density $p(\mathbf{x}_n|\mathbf{y}_n)$. Optimal localization is then achieved by minimizing (6).

5 Model description

To minimize the empirical risk (6) we need models for the densities $p(\mathbf{x},\mathbf{y})$ and $p(\mathbf{y})$. An attractive choice that makes no particular assumptions about the statistical nature of the observations is through *kernel smoothing* [13, 9]. A spherical multivariate Gaussian kernel of width h is centered on each training vector $(\mathbf{x}_n, \mathbf{y}_n)$, giving rise to the approximations

$$p(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{n=1}^{N} K_2(\mathbf{x}-\mathbf{x}_n)K_q(\mathbf{y}-\mathbf{y}_n) \quad (7)$$

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} K_2(\mathbf{x}-\mathbf{x}_n) \quad (8)$$

$$p(\mathbf{y}) = \frac{1}{N}\sum_{n=1}^{N} K_q(\mathbf{y}-\mathbf{y}_n) \quad (9)$$

where

$$K_2(\mathbf{x}-\mathbf{x}_n) = \frac{1}{2\pi h^2} \exp\left(-\frac{||\mathbf{x}-\mathbf{x}_n||^2}{2h^2}\right) \quad (10)$$

$$K_q(\mathbf{y}-\mathbf{y}_n) = \frac{1}{(2\pi h^2)^{q/2}} \exp\left(-\frac{||\mathbf{y}-\mathbf{y}_n||^2}{2h^2}\right) \quad (11)$$

a bivariate and q-variate Gaussian kernel, respectively. The width h reflects the degree of smoothness or overfitting of the model and its value for a particular problem can be computed by, e.g., cross-validation techniques [13, 9]. Finally, the use of kernel smoothing for density estimation provides the model $p(\mathbf{y}|\mathbf{x})$ for localization as

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})} \quad (12)$$

with $p(\mathbf{x},\mathbf{y})$ and $p(\mathbf{x})$ from (7) and (8), respectively.

Although direct implementation of the risk quantity (6) using the above formulas is possible, its computational cost is $O(N^2)$, with N the size of the training set, which can be regarded infeasible for large data sets. However, a very efficient implementation through *binned* kernel density estimators and the Fast Fourier Transform drops the cost to $O(N)$ [10, 13].

6 Optimization

Using the estimates (7)–(11) and substituting the $\mathbf{y}$ from (2), the empirical risk (6) becomes a smooth function of the matrix $\mathbf{W}$ under the constraint (3) and can be minimized with constrained nonlinear optimization [2]. Alternatively we can follow an iterative approach. We first solve for a one-dimensional feature which minimizes the risk, then for a second one which is orthogonal to the first, and so on for a specific number of features or until the risk gets no significant decrease.

For one-dimensional projections $y = \mathbf{w}^T\mathbf{z}$ the only constraint that is imposed from (3) is that the norm of the projection vector $\mathbf{w}$ must be one. Thus the transformation

$$\tilde{\mathbf{w}} = \frac{\mathbf{w}}{||\mathbf{w}||} \tag{13}$$

and computation of the risk using $y = \tilde{\mathbf{w}}^T\mathbf{z}$ reduces the problem to an *un*constrained nonlinear optimization problem which can be solved by regular techniques [2]. To compute the gradient of the risk (6) with respect to the vector $\mathbf{w}$ we need the gradient of y which is

$$\nabla_{\mathbf{w}} y = \nabla_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{z}}{||\mathbf{w}||} = \frac{\mathbf{z}^T - (\tilde{\mathbf{w}}^T\mathbf{z})\tilde{\mathbf{w}}^T}{||\mathbf{w}||}. \tag{14}$$

However, care must be taken when using gradient-based optimization because due to the nonlinearities of the risk (6) and the kernels (10) and (11) the objective function may easily get stuck in local minima.

6.1 Canonical correlation analysis

One way to obtain a good starting solution for the optimization routine is through *canonical correlation analysis* (CCA) [6]. This statistical method seeks linear transformations of two data sets so that the correlation between transformed variables from different sets is maximized. In our case we apply CCA between the $\mathbf{z}$ points and each coordinate of $\mathbf{x} = [x_1, x_2]$ separately to find two projection directions and then choose the direction with the smallest risk.

Formally, CCA finds an optimal projection vector $\mathbf{w}_i$ so that the correlation coefficient between the projected variable $y = \mathbf{w}_i^T\mathbf{z}$ and the i-th coordinate x_i of $\mathbf{x}$ ($i = 1, 2$) is maximized. For x_i of zero mean and unit variance and sphered $\mathbf{z}$ the CCA optimal solution becomes [6]

$$\mathbf{w}_i = E[x_i\mathbf{z}] \tag{15}$$

which, in effect, maximizes the effectiveness of $\mathbf{z}$ in predicting x_i. The vector $\mathbf{w}_i$ that gives the smallest risk for $i = 1, 2$ can be used as initial guess in the optimization routine.[4]

6.2 Extracting more features

Assume at some iteration that r feature vectors have been extracted forming the columns of the projection matrix $\mathbf{W}$. This matrix forms an r-dimensional basis in the $\mathbf{z}$ space and we can find its orthogonal complement $\mathbf{W}_\perp$ by orthogonalizing $\mathbf{W}$ (e.g., with Gram-Schmidt) so that

$$\mathbf{W}^T\mathbf{W}_\perp = \mathbf{0}. \tag{18}$$

Then we project the data points $\mathbf{z}$ on the complementary subspace by multiplying them with $\mathbf{W}_\perp$ and apply the one-dimensional optimization procedure described above to get an optimal feature $\mathbf{w}_\perp$ in this space. The resulting vector is transformed back to the original space by $\mathbf{w} = \mathbf{W}_\perp^T\mathbf{w}_\perp$ and is orthogonal to all other features (columns) of $\mathbf{W}$. Moreover, the constraint (3) ensures that the new projected variable $y = \mathbf{w}^T\mathbf{z}$ is uncorrelated with all other projected variables (components) of $\mathbf{y}$. We can iteratively apply this procedure for a specific number of features or until the risk gets no significant decrease.

7 Experiments

For our experiments we used the MEMORABLE robot database. This database is provided by the Tsukuba Research Center, Japan, for the Real World Computing Project, and contains a supervised set of about 8000 robot positions and associated measurements of sonars, infrared sensors, and images from an omnidirectional camera (Fig. 2), taken by a Nomad mobile robot moving in a typical office environment (Fig. 3).

To check our method we used a subset of 300 images obtained randomly from the whole environment. The omnidirectional images were first transformed to 64 × 256 pixels panoramic images and then sphered through PCA to 299-d. The $\mathbf{x}$ data were normalized to unit variance per dimension. The optimization

[4] The use of CCA to obtain a good first solution to the optimization problem can also be justified by the following consideration. Assume a one-dimensional projected variable y and a one-dimensional x and expand the conditional entropy (5) as

$$H(x|y) = H(y|x) + H(x) - H(y). \tag{16}$$

The first term reflects the noise after the projection while the second and third terms are the entropies of x and y, respectively. Let us then try to find the best mapping $y = f(x)$ that minimizes (16). In the low noise limit the first term can be ignored while for the term $H(y)$ we can write [8, p. 565]

$$H(y) \leq H(x) + E_x[\log|f'(x)|] \tag{17}$$

with equality if f has a unique inverse. This implies a way of minimizing (16) by seeking invertible mappings f. One such solution is, e.g., the linear mapping, and maximization of the correlation coefficients through CCA corresponds to x–y dependences that are as linear as possible.

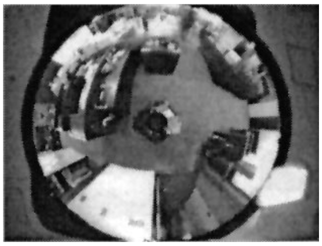

Figure 2: A snapshot of the omnidirectional camera.

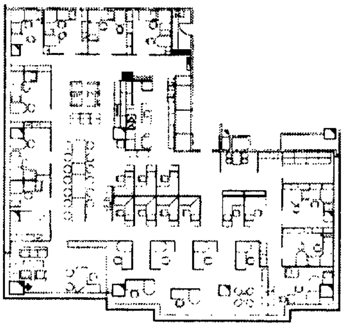

Figure 3: The office environment.

method described in the previous sections was used to extract the optimal features.

To compute the first feature we applied CCA between the $\mathbf{z}$ points and each individual coordinate x_1 and x_2 of $\mathbf{x}$ to get two solutions $\mathbf{w}_1$ and $\mathbf{w}_2$, respectively. It appeared that $\mathbf{w}_1$ had the smallest risk and this vector was used as initial guess in the nonlinear optimization routine. In the left part of Fig. 4 we show the projections of the $\mathbf{z}$ points on the first linear feature as a function of the $\mathbf{x}$ coordinates after nonlinear optimization. To compare with results from our previous research on PCA we show in the right part the projection of the data on the first principal component. We note that in the first case there is an almost linear relationship between the projected variable and the robot positions, a natural indication of good localization performance.

We iterated the procedure to compute the first 10 features. In Fig. 5 we show the risk as a function of the number of features (keeping the kernel width constant) for the proposed method and for PCA. We see that for small number of features our method outperforms PCA while for larger number of features the two methods converge. For nonlinear optimization we used the BFGS algorithm [2] while for the kernels width we empirically found that $h = N^{-2/7}$ gives good generalization performance.

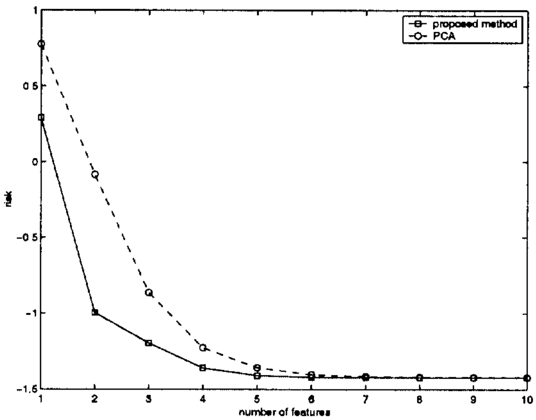

Figure 5: The risk as a function of the number of features.

8 Conclusions-discussion

We proposed a method for supervised linear feature extraction for robot localization. The main idea is the use of a supervised projection scheme that optimizes an appropriate localization criterion, a sort of *supervised* projection pursuit with a localization index [4]. The use of nonparametric density estimation provides a smooth objective function which can be optimized with respect to the projection matrix with nonlinear optimization, while canonical correlation analysis provides good starting solutions to the optimization routine. As the experiments indicate the method outperforms PCA as a linear feature extraction method for small number of features.

A similar approach to the problem of optimal feature extraction was proposed in [11] where a neural network was used to extract nonlinear features from images. In principle, nonlinear feature extraction can provide more relevant features for localization than a linear feature extraction method, however issues like model selection (e.g., deciding on the architecture of the neural network) or overfitting are more difficult to deal with than in the linear case.

Our method compares favorably to the method in [11] since it uses an objective function which can be computed in $O(N)$, with N the size of the training set, in contrast to the objective function in [11] which is $O(N^3)$. Moreover, canonical correlation analysis provides good initial solutions to the optimization problem possibly avoiding the local minima of the objective function.

In our analysis we assumed a constant value of the kernels width h and focussed on the optimal projec-

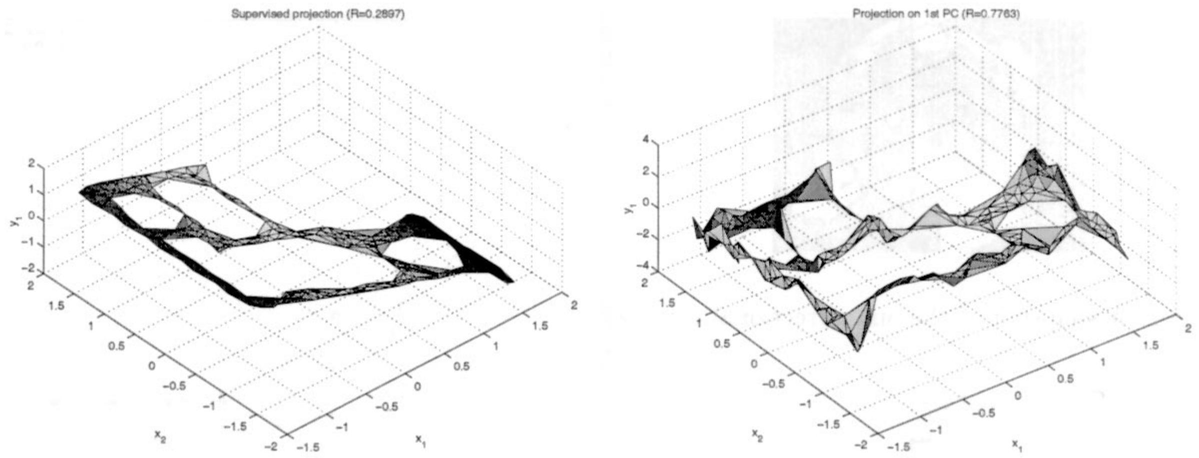

Figure 4: Projected observations on the first linear feature as a function of $\mathbf{x}$. (Left) Supervised projection: $R = 0.2897$. (Right) Projection on the first principal component: $R = 0.7763$.

tion matrix $\mathbf{W}$. An interesting problem which we currently investigate is the estimation from the data, using cross-validation techniques, of the optimal h and optimal number of features that lead to good generalization, especially when the size of the training set is small. Finally, recent experiments showed that constraint nonlinear optimization can provide better solutions than iterative optimization; these results together with mathematical details will appear elsewhere.

Acknowledgments

We would like to thank the people in the Real World Computing Project and the Tsukuba Research Center in Japan for providing us with the MEMORABLE robot database. Many thanks also to Roland Bunschoten for valuable discussions and various implementations.

References

[1] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd, Wellesley, MA, 1996.

[2] P. E. Gill, W. Murray, and M. Wright. *Practical Optimization*. Academic Press, London, 1981.

[3] S. Haykin. *Neural Networks*. Macmillan College Publishing Company, New York, 1994.

[4] M. C. Jones and R. Sibson. What is projection pursuit? *Journal of the Royal Statistical Society A*, 150:1–36, 1987.

[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] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

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

[8] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 3rd edition, 1991.

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

[10] B. W. Silverman. Kernel density estimation using the Fast Fourier Transform. *Appl. Statist.*, 31:93–99, 1982.

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

[12] N. Vlassis and B. Kröse. Robot environment modeling via principal component regression. In *Proc. IROS'99, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 677–682, Kyŏngju, Korea, Oct. 1999.

[13] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman & Hall, London, 1995.