Learning task-relevant features from robot data

Nikos Vlassis Roland Bunschoten Ben Kröse

RWCP, Autonomous Learning Functions SNN
Computer Science Institute
Faculty of Science
University of Amsterdam
The Netherlands
{vlassis,bunschot,krose}@science.uva.nl
http://www.science.uva.nl/research/ias

Abstract

Feature extraction from robot sensor data is a standard way to deal with the high dimensionality and redundancy of such data. An automatic, commonly used way to learn such features from a set of robot observations is Principal Component Analysis (PCA). However, as we argued in previous work, PCA can yield features with little discriminatory power between robot positions, leading to suboptimal localization performance of the robot. In order to get optimal task-relevant features, PCA must be replaced by a supervised projection method.

In this paper we extend our previously proposed supervised linear feature extraction method in two ways: (i) the projection matrix is optimized simultaneously over all columns under the constraint of orthonormality, (ii) a Jacobi parametrization of the matrix allows the use of unconstrained nonlinear optimization algorithms. The new algorithm is more efficient and many times faster than the old version. We show experimental results in extracting features from panoramic images of a mobile robot. The results compare favorably to the PCA solutions.

1 Introduction

In several mobile robot applications where a model of the environment must be built and used for navigation, appropriate *landmarks* or *features* must be extracted from the raw robot sensor measurements prior to modeling. The rationale is that normally the dimensionality of these data is very high, making any statistical inference in the original space unrealistic.

The features that are extracted from robot sensor data can be classified as *local* or *global*. The former usually refer to location-dependent distinctive characteristics of the environment like doors, hallways, etc., (natural landmarks), or landmarks realized through specialized devices like beacons (artificial landmarks) [1]. On the other hand, a global feature is normally location-independent and aims at providing good robot localization on the average.

Recently there has been a growing interest in automatic procedures that *learn* such features from a set of data (see, e.g., [13]). Automatic learning of features is a natural objective because on the one hand it obviates the need for man interference in the feature extraction process, while on the other hand makes the process (potentially) environment independent.

Learning features from a set of robot observations is most often carried out with statistical methods, and the easiest and most commonly used is Principal Component Analysis (PCA) [10]. This is a global feature extraction method which projects a set of robot observations linearly to a low-dimensional subspace, computed by solving a matrix eigenvalue problem. The nice thing about PCA is that it combines many optimality properties and is very simple to implement [10]. Recent reports on the use of PCA on mobile robots are [8, 2, 6, 11, 15, 5].

However, when the robot observations are collected in a 'supervised' manner, i.e., when they are annotated in the sample with the position of the robot where each observation was taken, then, as argued in [16], PCA can be suboptimal. The reason is that PCA is an unsupervised feature extraction method that uses only the observed sensor vectors to compute the projection

directions, and thus the extracted features can have little discriminatory power between robot positions. If feature extraction is to be used for tasks like robot localization and navigation, then PCA should be substituted by a *supervised* projection method [16].

In the current paper we extend the results in [16] in two main ways. First, in the above work the projection directions were learned in a greedy fashion, namely, a projection to an optimal direction was computed, then a second optimal direction was sought which was orthogonal to the first, etc. This strategy can be suboptimal and it is not difficult to devise artificial data sets that show this suboptimal behavior. In this paper we optimize the projection matrix (see below) simultaneously for all dimensions while keeping its columns pairwise orthonormal.

Second, we adopt an optimization strategy which obviates the need for constrained nonlinear optimization by parametrizing the projection matrix as a product of Jacobi matrices satisfying the orthogonality constraint during optimization. These two improvements make the method more efficient and much faster than the original version.

In the following we first describe the proposed method and then show experimental results from its application in panoramic image data collected by a mobile robot in a typical indoor environment. The average localization performance—evaluated through an appropriate risk function—when using the proposed method vs. PCA, and the visualization of the projected data manifold in the reduced subspace permit a quantitative and qualitative verification of our theoretical claims.

2 Feature extraction and the localization risk

For clarity of exposition and visualization we will limit our analysis to a robot that follows a predefined one-dimensional trajectory in its workspace. The results extend directly to the general case. For each position (offset) s of the robot on the trajectory we assume that the sensors provide an observation vector $\mathbf{x} \in \mathbb{R}^{\mathrm{d}}$. For our analysis we assume a supervised training set $\{s_i, \mathbf{x}_i\}$, $1 \leq i \leq n$, of observations $\mathbf{x}_i$ collected at respective trajectory positions s_i.

Linear feature extraction amounts to reducing the dimensionality of the data $\mathbf{x}_i$ by linearly projecting them to a subspace $\mathbb{R}^{\mathrm{q}}$, $1 < q < d$, multiplying them with a $d \times q$ matrix $\mathbf{W}$ with orthonormal columns

$$\mathbf{y}_i = \mathbf{W}^T \mathbf{x}_i, \quad 1 \leq i \leq n, \quad \mathbf{W}^T \mathbf{W} = \mathbf{I}_q \qquad (1)$$

where $\mathbf{I}_q$ stands for the q-dimensional identity matrix. Moreover, we assume a probabilistic model that associates robot locations with sensor observations. For an observation $\mathbf{x}$ that is projected through (1) to a feature vector $\mathbf{y}$ we assume a model for $p(s|\mathbf{y})$, the conditional density of the robot position s given $\mathbf{y}$.

To assess the quality of an individual projection we must define an appropriate *risk* function that measures the average localization performance of the robot using the extracted features $\mathbf{y}_i$. For this purpose it was proposed in [13] the risk function

$$R_L = \frac{1}{n} \sum_{i=1}^{n} \int |s - s_i| p(s|\mathbf{y}_i) ds, \qquad (2)$$

i.e., the average over the training set mean absolute distance to the true—conditioned on the feature vector $\mathbf{y}_i$—location s_i. This risk penalizes position estimates that appear on the average far from the true position of the robot. The above formula was approximated in [13] from the training set with complexity $O(n^3)$.

In [16] we proposed an alternative risk which is $O(n^2)$. This risk is based on the simple observation that, for a given observation $\mathbf{x}_i$ which is projected through (1) to $\mathbf{y}_i$, the density $p(s|\mathbf{y}_i)$ will always exhibit a *mode* on $s = s_i$. Thus, an approximate measure of divergence from this mode is the Kullback-Leibler distance between $p(s|\mathbf{y}_i)$ and a unimodal density sharply peaked at $s = s_i$, giving the approximate estimate $-\log p(s_i|\mathbf{y}_i)$ plus a constant. Averaging over all points $\mathbf{y}_i$ we have to minimize the risk

$$R_K = -\frac{1}{n} \sum_{i=1}^{n} \log p(s_i|\mathbf{y}_i) \qquad (3)$$

which can be regarded as the average negative log-likelihood of the data given the model of $p(s_i|\mathbf{y}_i)$ and the projection matrix $\mathbf{W}$.

From (3) we see that a nonparametric estimate of $p(s|\mathbf{y})$ is needed. For an appropriate sequence of weights $\lambda_j(\mathbf{y}), 1 \leq j \leq n$, such an estimate is [12]

$$p(s|\mathbf{y}) = \sum_{j=1}^{n} \lambda_j(\mathbf{y}) \phi_{h_s}(s - s_j) \qquad (4)$$

where

$$\phi_{h_s}(s) = \frac{1}{\sqrt{2\pi} h_s} \exp\left(-\frac{s^2}{2h_s^2}\right) \qquad (5)$$

is the univariate Gaussian kernel with bandwidth h_s, defining a local *smoothing* region around s. A weight function $\lambda_j(\mathbf{y})$ which satisfies the conditions in [12] and makes the above estimate a smooth function of the projection matrix $\mathbf{W}$ is

$$\lambda_j(\mathbf{y}) = \frac{\phi_{h_y}(\mathbf{y}-\mathbf{y}_j)}{\sum_{k=1}^{n}\phi_{h_y}(\mathbf{y}-\mathbf{y}_k)} \tag{6}$$

where

$$\phi_{h_y}(\mathbf{y}) = \frac{1}{(2\pi)^{q/2}h_y^q}\exp\left(-\frac{||\mathbf{y}||^2}{2h_y^2}\right) \tag{7}$$

is the q-dimensional spherical Gaussian kernel with bandwidth h_y. The two kernel bandwidths h_y and h_s are the only free parameters of the model $p(s|\mathbf{y})$ and their values affect the resulting projections. Substituting $p(s|\mathbf{y})$ from above into (3) we get a risk with complexity $O(n^2)$.

3 Model selection and optimization

3.1 Kernel smoothing

Using a nonparametric estimate of a density using (4) and (5)–(7) requires a choice for the smoothing parameters y_s and h_y. Our approach was to assign constant values to these two bandwidths during optimization. For projections to 2-d we set $h_y = n^{-2/7}$ which can be kept fixed during optimization after sphering the data (see next). This value is within the optimal bounds $O(n^{-1/3})$ and $O(n^{-1/4})$ given in [4, Sec. 4] for the related problem of projection pursuit regression, while it was found to give good results in practice. For the s-bandwidth we chose the Gaussian MISE optimal value $h_s = (3n/4)^{-1/5}$ [17, Ch. 3.2].

3.2 Sphering

A sphering of the data $\mathbf{x}_i$, namely, a normalization to zero mean and identity covariance matrix, makes the kernel bandwidth h_y independent of the projection. Then h_y can be kept constant during optimization leading to considerable computational savings. Sphering means a rotation of the data to their PCA directions and then standardization of the individual variances to one. To avoid modeling noise in the data, it is typical to ignore directions with small eigenvalues, and a heuristic method to do this is by putting a threshold to the ratio of the cumulative variance (added eigenvalues) to the total variance.

The numerically most accurate way to sphere the data is by singular value decomposition [9]. Let $\mathbf{X}$ be the $n \times d$ matrix whose rows are the data $\mathbf{x}_i$ after they have been normalized to zero mean. For $n > d$, we compute the singular value decomposition $\mathbf{X} = \mathbf{ULV}^T$ of the matrix $\mathbf{X}$ and form the matrix $\mathbf{A} = \sqrt{n}\mathbf{VL}^{-1}$. The points $\mathbf{XA}$ are then sphered [10].

For $n \leq d$ the data $\mathbf{x}_i$ lie in general in a $(n-1)$-dimensional Euclidean subspace of $\mathbb{R}^d$. In this case it is more convenient to compute the principal directions through eigenanalysis of $\mathbf{K} = \mathbf{XX}^T$, the inner products matrix of the zero mean data. We compute its singular value decomposition $\mathbf{K} = \mathbf{ULV}^T$ and remove the last column of $\mathbf{V}$ and last column and row of $\mathbf{L}$ (the last eigenvalue of $\mathbf{K}$ will always be zero). Then we form the matrix $\mathbf{A} = \sqrt{n}\mathbf{VL}^{-1}$. The points $\mathbf{KA}$ are $(n-1)$-dimensional and sphered [7].

Moreover, all projections of sphered data $\mathbf{x}_i$ in the form of (1) give also sphered data $\mathbf{y}_i$ because

$$E[\mathbf{yy}^T] = \mathbf{W}^T E[\mathbf{xx}^T]\mathbf{W} = \mathbf{I}_q \tag{8}$$

due to the constraint of orthonormal columns of $\mathbf{W}$. This frees us from having to reestimate (co)variances of the projected data in each step of the optimization algorithm. In the following we assume that the data $\mathbf{x}_i$ have already been sphered and the position data s_i have been normalized to zero mean and unit variance.

3.3 Optimization

The smooth form of the risk R_K as a function of $\mathbf{W}$ allows the minimization of the former with nonlinear optimization. For constrained optimization we must compute the gradient of R_K and the gradient of the constraint function $\mathbf{W}^T\mathbf{W}-\mathbf{I}_q$ with respect to $\mathbf{W}$, and then plug these estimates in a constrained nonlinear optimization routine to optimize with respect to R_K [3].

An alternative approach which avoids the use of constrained nonlinear optimization, in a similar problem using kernel smoothing for discriminant analysis, has been recently proposed in [14]. The idea is to parametrize the projection matrix $\mathbf{W}$ by a product of *Jacobi* rotation matrices [9] and then optimize with respect to the angle parameters involved in each matrix. For projections from $\mathbb{R}^d$ to $\mathbb{R}^q$ this parametrization takes the form

$$\mathbf{W} = \prod_{o=1}^{q}\prod_{u=q+1}^{d}\mathbf{G}_{ou} \tag{9}$$

where $\mathbf{G}_{ou}$ is a Jacobi rotation matrix which equals $\mathbf{I}_d$ except for the elements $g_{oo} = \cos\theta_{ou}$, $g_{ou} = \sin\theta_{ou}$,

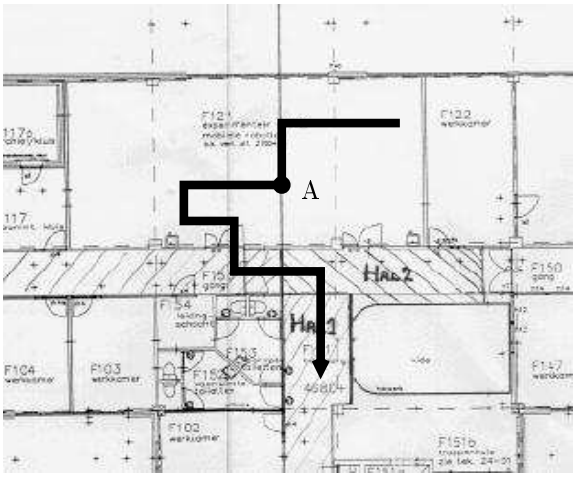


Figure 1: The robot trajectory.

$g_{uo} = -\sin\theta_{ou}$, and $g_{uu} = \cos\theta_{ou}$ for an angle θ_{ou} which depends on o and u. For simplicity we let in the above notation g_{oo}, g_{ou}, etc., denote the (o,o)-th, (o,u)-th, etc., elements of the matrix $\mathbf{G}_{ou}$, respectively. To ensure that $\mathbf{W}$ is $d \times q$, only the first q columns of the last matrix $\mathbf{G}_{qd}$ in (9) are retained, while multiplications must be carried from right to left to reduce the evaluation cost.

Multiplication with a matrix $\mathbf{G}_{ou}$ causes a rotation by θ_{ou} along the plane defined by the dimensions o and u, while the range of indices in (9) ensures that all rotations take place along planes defined by at least one non-projective direction, i.e., one among the $d - p$ remaining dimensions. This fact also reduces the total number of parameters from qd in the constrained optimization case (elements of matrix $\mathbf{W}$) to $q(d-q)$ here (angles θ_{ou}).

The derivative of the risk R_K with respect to an angle θ_{kl} is (we skip an analytical derivation here)

$$\frac{\partial}{\partial\theta_{kl}} R_K = \mathrm{trace}\left\{ (\nabla_{\mathbf{W}} R_K)^T \left(\frac{\partial}{\partial\theta_{kl}} \mathbf{W} \right) \right\} \qquad (10)$$

where the first term in the trace is

$$\nabla_{\mathbf{W}} R_K = \frac{1}{n h_y^2} \mathbf{X}^T [\mathbf{B} + \mathbf{B}^T - \mathrm{diag}(\mathbf{1}^T \mathbf{B})] \mathbf{X}\mathbf{W} \qquad (11)$$

where $\mathbf{X}$ is the $n \times d$ matrix of the sphered data, $\mathbf{1}$ is a column vector of all ones, $\mathrm{diag}(\cdot)$ transforms a vector to a diagonal matrix, and $\mathbf{B}$ is the $n \times n$ matrix with elements

$$b_{ij} = \lambda_j(\mathbf{y}_i) - \frac{\phi_{h_y}(\mathbf{y}_i - \mathbf{y}_j)\phi_{h_s}(s_i - s_j)}{\sum_{k=1}^{n} \phi_{h_y}(\mathbf{y}_i - \mathbf{y}_k)\phi_{h_s}(s_i - s_k)}. \qquad (12)$$

Figure 2: Panoramic snapshot from position A.

The second term of the trace is

$$\frac{\partial}{\partial\theta_{kl}} \mathbf{W} = \prod_{o=1}^{q} \prod_{u=q+1}^{d} \frac{\partial}{\partial\theta_{kl}} \mathbf{G}_{ou} \qquad (13)$$

where

$$\frac{\partial}{\partial\theta_{kl}} \mathbf{G}_{ou} = \begin{cases} \mathbf{G}'_{ou} & \text{if } k = o \text{ and } l = u \\ \mathbf{G}_{ou} & \text{otherwise} \end{cases} \qquad (14)$$

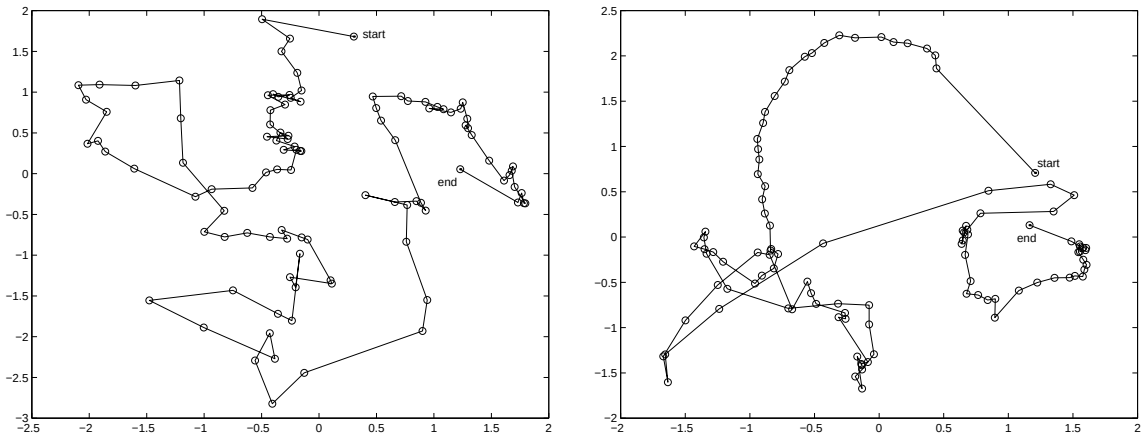
and $\mathbf{G}'_{ou}$ is the matrix $\mathbf{G}_{ou}$ with the ones substituted by zeros and the trigonometric functions substituted by their derivatives.

A point we should note is that the mixture density form of (4) and the additional trigonometric functions in (9) can make the landscape of the risk R_K have numerous local minima. For this reason, combining a gradient-free optimization method like, e.g., Nelder-Mead [9], with nonlinear optimization is requisite. Also an appropriate dimension reduction through sphering prior to optimization can significantly facilitate the search. In any case, the optimization algorithm must be applied many times and the solution with the minimum risk must be retained.

4 Experiments

We applied the above algorithm to data collected by a Nomad Scout robot following a predefined trajectory in our mobile robot lab and the adjoining hall as shown in Fig. 1. The omnidirectional imaging device which is mounted on top of the robot consists of a vertically mounted standard camera aimed upward looking into a spherical mirror. The data set contains 104 omnidirectional images (320×240 pixels) captured every 25 centimeters along the robot path. Each image is transformed to a panoramic image (64×256) and this set of 104 panoramic images constitutes the training set of our algorithm. A typical panoramic image shot at the position A of the trajectory is shown in Fig. 2.

In order to apply our supervised projection method, we first sphered the panoramic image data using the



Proposed method: $R = -85$ PCA: $R = -72$

Figure 3: Projection of the sphered panoramic image data from 10-d to 2-d: using the proposed method (left), projection on the first two principal components (right). The 'start' and 'end' points are the projections of the panoramic images captured by the robot at the beginning and end, respectively, of its trajectory.

inner products matrix as explained above and kept the first 10 dimensions explaining about 60% of the total variance. Then we applied our method projecting the sphered data points from 10-d to 2-d. The resulting two-dimensional points are shown on the left part of Fig. 3. For optimization we ran several times a combined search using the Nelder-Mead algorithm with random initial values for the Jacobi angles in $[-\pi/2, \pi/2]$, together with nonlinear optimization with the BFGS algorithm [3, 9]. Running only BFGS required many more runs with random initial guesses to reach the global minimum, leading to comparable total expenses. Each execution of the optimization algorithm took a couple of seconds in a Sparc Ultra 5 machine.

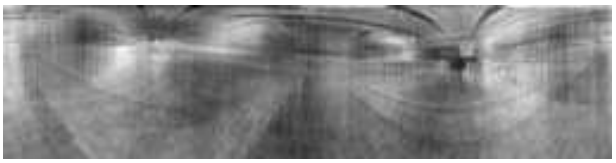
On the right part of Fig. 3 we show the result of projecting the sphered 10-d points on the first two principal components of the data. We clearly see the advantage of the proposed method over PCA. The risk is smaller, while from the shape of the projected manifold we see that taking into account the pose information during projection can significantly improve the resulting features: there are fewer self-intersections of the projected manifold in our method than in PCA which, in turn, means better robot position estimation on the average.

Finally, in Fig. 4 we show the first two feature vectors (points in the original space of panoramic images) learned by our method and by PCA. In the PCA case these are the familiar first two eigenimages of the panoramic data which, as is normally observed in typical data sets, exhibit low spatial frequencies. We see that the proposed supervised projection method yields very different feature vectors than PCA, namely, images with higher spatial frequencies and distinct characteristics.

5 Conclusions

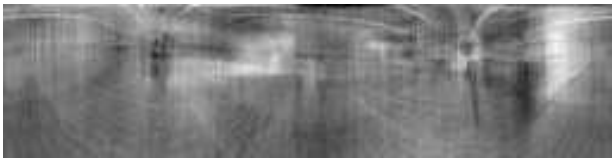
We proposed a method for learning task-relevant linear features from high-dimensional robot observations. Our method is supervised in the sense that the position of the robot in the sample is also taken into account during optimization. This makes the method superior to PCA which is unsupervised. We showed results of linear feature extraction from panoramic robot data when the robot was moving in a typical office environment. The results show clearly the superiority of the proposed method over PCA.

Our method can be useful in various robotic settings and is not limited to mobile robots. In particular, it can used in any case where global feature extraction from *supervised* robot observations is in order. The extension of the method to handle nonlinear features is possible (e.g., by using a neural network) but then additional issues have to be addressed (complexity of the network, overfitting, etc.). Besides, the wide use of PCA in robotic problems shows that linear feature extraction is still a viable approach in robotics.



1st optimal feature vector

1st eigenvector



2nd optimal feature vector

2nd eigenvector

Figure 4: The first two feature vectors using our method (left), and PCA (right).

References

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

[2] 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.

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

[4] P. Hall. On projection pursuit regression. *Ann. Statist.*, 17(2):573–588, 1989.

[5] M. Jogan and A. Leonardis. Robust localization using the eigenspace of spinning-images. In *Proc. IEEE Workshop on Omnidirectional Vision*, South Carolina, June 2000.

[6] B. Kröse and R. Bunschoten. Probabilistic localization by appearance models and active vision. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2255–2260, Detroit, Michigan, May 1999.

[7] V. Kumar and H. Murakami. Efficient calculation of primary images from a set of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 4(5):511–515, 1982.

[8] S. K. Nayar, H. Murase, and S. A. Nene. Learning, positioning, and tracking visual appearance. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3237–3244, San Diego, CA, 1994.

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

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

[11] R. Sim and G. Dudek. Learning visual landmarks for pose estimation. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, May 1999.

[12] C. J. Stone. Consistent nonparametric regression (with discussion). *Ann. Statist.*, 5:595–645, 1977.

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

[14] K. Torkkola and W. Campbell. Mutual information in learning feature transformations. In *Proc. Int. Conf. on Machine Learning*, Stanford, CA, June 2000.

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

[16] N. Vlassis, Y. Motomura, and B. Kröse. Supervised linear feature extraction for mobile robot localization. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2979–2984, San Fransisco, CA, Apr. 2000.

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