A probabilistic model for appearance-based robot localization

B.J.A Kröse, N. Vlassis, R. Bunschoten,

Real World Computing Partnership, Novel Functions Laboratory SNN
Department of Computer Science, University of Amsterdam
Kruislaan 403, NL-1098 SJ Amsterdam, The Netherlands
{krose,vlassis,bunschot}@science.uva.nl

Y. Motomura,

Electrotechnical Laboratory, Umezono 1-1-4, Tsukuba Ibaraki 305-8568, JAPAN
motomura@etl.go.jp

Abstract

In this paper we present a method for an appearance based modeling of the environment of a mobile robot. We describe the task (localization of the robot) in a probabilistic framework. Linear image features are extracted using a Principal Component Analysis. The appearance model is represented as a probability density function of the image feature vector given the location of the robot. We estimate this density model from the data with a kernel estimation method. We show how the parameters of the model influence the localization performance. We also study how many features and which features are needed for good localization.

Key words: Robot localization, Feature extraction, Probabilistic modeling

1 Introduction

An internal representation of the environment is needed to navigate a mobile robot optimally from a current state toward a desired state. In the traditional state of affairs such a model is represented as a geometric model in the task space of the robot. A path to the desired location can be planned which has to be followed by the robot. Path following requires a good estimation of the location of the robot. The proprioceptive sensors (shaft encoders, which measure the revolutions of the wheels) are subject to inaccuracies as a result of slip. Therefore, external sensors are needed to update the position estimate.

Preprint submitted to Elsevier Preprint *23 October 2000*

Because sensor signals are noisy, a wide variety of probabilistic methods have been developed to obtain a robust estimate of the location of the robot given its sensory inputs. These methods generally incorporate some observation model which gives the probability of the sensor measurement given the location of the robot and the parameterization of the environment (the *map*). Sometimes this parameter vector describes explicit properties of the environment (such as positions of landmarks [20] or occupancy values [8]) or describes an implicit relation between sensor pattern and location (such as neural networks [13], radial basis functions [21] or look-up tables [2]).

In our laboratory we use a vision sensor for robot localization. One possible approach for modeling the environment is to make a full 3D CAD model of the environment [18] to localize the robot. However, such a model requires the estimation of many parameters. Recently, in the field of object recognition, methods have been proposed which model the data directly in the visual domain, instead of making a geometric model from a set of visual observations (appearance modeling [12]). In this paper we describe such an approach for environment modeling of a mobile robot.

Modeling the relation between robot location and visual observation is not a trivial task, particularly since the images are high-dimensional data vectors. For an accurate modeling of these data we should need an extremely large set of learning points. It is therefore essential that the dimensionality of the data be reduced adequately before the modeling step. Feature extraction becomes increasingly 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 projection of high- dimensional data onto low dimensional subspaces. For appearance modeling of objects, a linear feature set which is obtained from a Principal Component Analysis (PCA) of the data has been used successfully [12]. In a number of robotic applications PCA has been proposed for finding linear features from image data. A number of authors [10,14,9] describe PCA features for the localization of a mobile robot and others use PCA features for visual servoing [3] or image synthesis of moving robot manipulators [6]. For dense range sensor scans PCA has been used to decrease the dimensionality [2,23]. PCA exhibits many optimality properties, e.g., uncorrelated features, maximum variance, minimum reconstruction error, etc., but needs not be the best feature extraction method for the task at hand (localization or visual servoing). Nonlinear features for robot navigation have been proposed as well, such as landmark-based approaches [11] [16], or neural networks which operate on a number of predefined image features as color, etc [19], or local autocorrelation functions [5]. In this paper we will use PCA features.

In the next section we will first describe the general framework of our probabilistic localization procedure. In section 3 we describe the PCA algorithm and in section 4 we describe our approach for learning the observation model from a set of training data. In section 5 we discuss our criteria which can be used for the evaluation of the model and the localization procedure. In section 6 we describe an experiment in which we investigate the accuracy of the model.

2 Probabilistic appearance-based robot localization

Let $\mathbf{x}$ be a stochastic vector (e.g., 2-D or 3-D) denoting the robot position in the workspace $\mathcal{X}$. Similar to [1] we employ a form of *Markov localization* for our mobile robot. This means that at each point in time we have a belief where the robot is indicated by a probability density $p(\mathbf{x})$. Markov localization requires two probabilistic models to maintain a good position estimate: a *motion* model and an *observation* model.

The motion model describes the effect which a motion command has on the location of the robot and can be represented by a conditional probability density

$$p(\mathbf{x}_t|u, \mathbf{x}_{t-1}) \tag{1}$$

which determines the distribution of $\mathbf{x}_t$ (the position of the robot after the motion command u) if the initial robot position is $\mathbf{x}_{t-1}$.

The observation model describes the relation between the observation, the location of the robot, and the parameters of the environment. In our situation the robot takes an omnidirectional image $\mathbf{z}$ at position $\mathbf{x}$. We consider this as a realization of a stochastic variable $\mathbf{z}$. The observation model is now given by the conditional distribution

$$p(\mathbf{z}|\mathbf{x}; \boldsymbol{\theta}), \tag{2}$$

in which the parameter vector $\boldsymbol{\theta}$ describes the distribution and reflects the underlying 'environment' model.

Using the Bayes' rule we can get an estimate of the position of the robot after observing $\mathbf{z}$:

$$p(\mathbf{x}|\mathbf{z}; \boldsymbol{\theta}) = \frac{p(\mathbf{z}|\mathbf{x}; \boldsymbol{\theta})p(\mathbf{x})}{\int p(\mathbf{z}|\mathbf{x}; \boldsymbol{\theta})p(\mathbf{x})d\mathbf{x}} \tag{3}$$

Here $p(\mathbf{x})$ gives the probability that the robot is at $\mathbf{x}$ before observing $\mathbf{z}$. Note that $p(\mathbf{x})$ can be derived using the old information and the motion model $p(\mathbf{x}_t|u, \mathbf{x}_{t-1})$ repeatedly. If both models are known we can combine them and decrease the motion uncertainty by having a new observation.

In this paper we will focus on the observation model. Often the parameter vector $\boldsymbol{\theta}$ is an explicit 'map' of the environment, such as an occupancy grid [20,8] or the positions of landmarks [19]. Such a map can be provided from prior knowledge, but here the key issue is to learn this model from data. The approach that we will adopt is that we are not going to estimate the parameters of some sort of CAD model, but our map will consist of an implicit model of the data. The problem of map building is now to find the most likely set of parameters $\boldsymbol{\theta}$ given a dataset consisting of positions $\mathbf{x}$ and corresponding observations $\mathbf{z}$ [1].

In order to estimate these parameters from data we need a huge amount of data, particularly since the dimensionality of $\mathbf{z}$ (in our case the omnidirectional images) is high. Therefore, as is usually done in appearance modeling, the dimensionality of the sensor data has to be reduced. Here we restrict ourselves to linear projections, in which the image can be described as a set of linear features.

3 Principal Component Analysis

Let us assume that we have a set of N images $\{\mathbf{z}_n\}$, $n = 1, \ldots, N$. The images are collected at respective 2-dimensional robot positions $\{\mathbf{x}_n\}$. Each image consists of d pixels and is considered as a d-dimensional data vector. In a Principal Component Analysis (PCA) the eigenvectors of an image set are computed and used as an orthogonal basis for representing individual images. Although, in general, for perfect reconstruction all eigenvectors are required, only a few are sufficient for visual recognition. These eigenvectors constitute the $q, (q < d)$ dimensions of the *eigenspace*. PCA projects the data onto this space in such a way that the projections of the original data are uncorrelated, while most of the variation of the original data set is preserved.

First we subtract from each image the average image over the entire image set, $\bar{\mathbf{z}}$. This ensures that the eigenvector with the largest eigenvalue represents

[1] In this paper we assume we have a set of positions and corresponding observations. It is also possible to do a concurrent localization and map building. In this case the only available data is a stream of data $\{\mathbf{z}^{(1)}, u^{(1)}, \mathbf{z}^{(2)}, u^{(2)}, \ldots, \mathbf{z}^{(T)}, u^{(T)}\}$ in which u is the motion command to the robot. Using a model about the uncertainty of the motion of the robot it is possible to estimate the parameters from these data [20].

the direction in which the variation in the set of images is maximal. We now stack the N image vectors to form the rows of an $N \times d$ image matrix $\mathbf{Z}$. The numerically most accurate way to compute the eigenvectors from the image set is by taking the singular value decomposition [15] $\mathbf{Z} = \mathbf{ULV}^T$ of the image matrix $\mathbf{Z}$, where $\mathbf{V}$ is a $d \times q$ orthonormal matrix with columns corresponding to the q eigenvectors $\mathbf{v}_i$ with largest eigenvalues λ_i of the covariance matrix of $\mathbf{Z}$ [7].

These eigenvectors $\mathbf{v}_i$ are now the linear features. Note that the eigenvectors are vectors in the d-dimensional space, and can be depicted as images: the *eigenimages*. The elements of the $N \times q$ matrix $\mathbf{Y} = \mathbf{ZV}$ are the projections of the original d-dimensional points to the new q-dimensional eigenspace $\mathcal{Y}$ and are the q-dimensional feature values.

4 Observation model

The linear projection gives us a feature vector $\mathbf{y}$, which we will use for localization. The Markov localization procedure, as presented in Section 2, is used on the feature vector $\mathbf{y}$:

$$p(\mathbf{x}|\mathbf{y};\boldsymbol{\theta}) \propto p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})p(\mathbf{x}), \tag{4}$$

We now have to find a method to estimate the observation model $p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$ from a dataset $\{\mathbf{x}_n, \mathbf{y}_n\}, n = 1, \ldots, N$. In earlier work [23] we modeled the conditional density for each feature independently as a univariate Gaussian centered on a parameterized mean $f(\mathbf{x})$ and having a parameterized variance $s(\mathbf{x})$:

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

where we used radial basis functions neural networks to represent $f(\mathbf{x})$ and $s(\mathbf{x})$. From such a model for the densities of the individual features we approximated the joint conditional density as the product of the marginal densities:

$$p(\mathbf{y}|\mathbf{x}) = \prod(p(y|\mathbf{x}).$$

Note that, although PCA ensures global independence between the features, local (input-dependent) correlations between features are ignored by such an approximation.

Instead of using radial basis function neural networks we also presented methods to model the distribution $p(y|\mathbf{x})$ with a mixture of Gaussians, using an EM method to estimate the parameters [21]. This works adequately for 1D problems, however if we want to model the distribution of the whole feature vector we need to estimate large covariance matrices.

In this paper we use a kernel density estimation or 'Parzen' estimator. In a Parzen approach the density function is approximated by a sum of kernel functions around the N data points from the training set. Note that in a strict sense this is not a 'parametric' technique in which the parameters of some pre-selected model are estimated from the training data. On the other hand, the training points themselves as well as the chosen kernel width may be considered as the parameter vector $\boldsymbol{\theta}$. We write $p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$ as

$$p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) = \frac{p(\mathbf{y},\mathbf{x};\boldsymbol{\theta})}{p(\mathbf{x};\boldsymbol{\theta})} \tag{5}$$

and represent each of these distribution as a sum of kernel functions:

$$p(\mathbf{x},\mathbf{y};\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N} g_y(\mathbf{y}-\mathbf{y}_n)g_x(\mathbf{x}-\mathbf{x}_n). \tag{6}$$

Once the form is adopted, the kernel function is not required to be uniform. A widely used implementation is a multivariate *Gaussian kernel*, with a kernel width determined by h:

$$g_y(\mathbf{y}-\mathbf{y}_n) = \frac{1}{(2\pi)^{q/2}h^q|\Sigma|^{1/2}}\exp[-\frac{1}{2h^2}(\mathbf{y}-\mathbf{y}_n)^T\Sigma^{-1}(\mathbf{y}-\mathbf{y}_n)], \tag{7}$$

in which q is the dimensionality of the feature space and

$$g_x(\mathbf{x}-\mathbf{x}_n) = \frac{1}{2\pi h^2|\Sigma|^{1/2}}\exp[-\frac{1}{2h^2}(\mathbf{x}-\mathbf{x}_n)^T\Sigma^{-1}(\mathbf{x}-\mathbf{x}_n)] \tag{8}$$

where h must satisfy $\lim_{N\to\infty} h^n = 0$, $\lim_{N\to\infty} Nh^n = \infty$ and $\lim_{N\to\infty} Nh^2n = \infty$ for asymptotic unbiasedness, consistency and uniform consistency [4].

Similarly,

$$p(\mathbf{x};\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N} g_x(\mathbf{x}-\mathbf{x}_n). \tag{9}$$

The difficulty with kernel methods is the choice of the covariance matrix (see [17]). In our experiments we used a unit covariance matrix and used the same

h for the $\mathbf{x}$ and $\mathbf{y}$ kernels.

The question is whether we should use all features for localization or whether we can use a subset of the features. Also, we would like to investigate whether the ranking of features which comes with the PCA, and gives the features which are most important for *reconstruction* of the image, gives also the optimal set of features for localization.

In order to proceed we first have to define an evaluation measure.

5 Evaluation of localization

In order to assess the performance of the model we need an evaluation measure. An obvious criterion for the task is the error between the true and the estimated position. Such a risk function for robot localization was recently proposed in [19]. Suppose the difference between the true position $\mathbf{x}^*$ of the robot and the the estimated position by $\mathbf{x}$ is denoted by the loss function $L(\mathbf{x}, \mathbf{x}^*)$. If the robot observes $\mathbf{y}^*$, the expected localization error $\varepsilon(\mathbf{x}^*, \mathbf{y}^*)$ is (using Bayes' rule 2) computed as

$$\begin{aligned} \varepsilon(\mathbf{x}^*, \mathbf{y}^*) &= \int_{\mathbf{x}} L(\mathbf{x}, \mathbf{x}^*) p(\mathbf{x}|\mathbf{y}^*) d\mathbf{x} \\ &= \int_{\mathbf{x}} L(\mathbf{x}, \mathbf{x}^*) \frac{p(\mathbf{y}^*|\mathbf{x}) p(\mathbf{x})}{p(\mathbf{y}^*)} d\mathbf{x} \end{aligned} \tag{10}$$

To obtain the total risk for the particular model, the above quantity must be averaged over all possible observations $\mathbf{y}^*$ obtained from $\mathbf{x}^*$ and all possible $\mathbf{x}^*$ to give

$$R_{loss} = \int_{\mathbf{x}^*} \int_{\mathbf{y}^*} \varepsilon(\mathbf{x}^*, \mathbf{y}^*) p(\mathbf{y}^*, \mathbf{x}^*) d\mathbf{y}^* d\mathbf{x}^* \tag{11}$$

The *empirical* risk is computed when estimating this function from the data:

$$\begin{aligned} R_{loss} &= \frac{1}{N} \sum_{n=1}^{N} \varepsilon(\mathbf{x}_n, \mathbf{y}_n) \\ &= \frac{1}{N} \sum_{n=1}^{N} \frac{\sum_{l=1}^{N} L(\mathbf{x}_l, \mathbf{x}_n) p(\mathbf{y}_n|\mathbf{x}_l)}{\sum_{l=1}^{N} p(\mathbf{y}_n|\mathbf{x}_l)}. \end{aligned} \tag{12}$$

Another approach for evaluating the contribution of a feature y_i is based on information theoretic considerations [22]. Denote the estimated position $\mathbf{x}$ as a random variable X and the measurements $\mathbf{y}$ as a random variable Y. The entropy of a random variable indicates the amount of information in a distribution. The conditional entropy $H(X|Y)$ gives the expected information in X if a measurement is taken. As a loss function we now define the *mutual information*: the difference between the entropy before and the entropy after the measurement:

$$\begin{aligned} R_{entropy} &= H(X) - H(X|Y) \\ &= H(X) + H(Y) - H(X,Y). \end{aligned} \tag{13}$$

The entropy of the positions $H(X)$ is not dependent on the features so we can skip this one, and use for the evaluation the approximation from the data:

$$R_{entropy} = -\sum_{n=1}^{N} \log p(\mathbf{y}_n) + \sum_{n=1}^{N} \log p(\mathbf{x}_n, \mathbf{y}_n) \tag{14}$$

It is easy to see that this risk is equal to the Kullback-Leibler divergence between a delta-peaked density on the real robot position $\mathbf{x}^*$ and the distribution $p(\mathbf{x}|\mathbf{y}^*)$, averaged over the whole training set. Thus, the mutual information provides an average localization criterion in itself, but has a lower computational complexity. In this paper we will give the evaluation of both methods.

6 Experiments on appearance based localization

In the previous sections we have described a framework for learning a probabilistic localization model from data. The model uses linear features derived from a vision system. In a series of experiments we want to investigate experimentally:

- How does the performance depend on the model parameters?
- How many features are needed, and
- Which linear features give the best performance?

6.1 Datasets and preprocessing

We tested our methods on real image data obtained from a robot moving in an office environment, of which an overview is shown in figure 3. We made use of the MEMORABLE robot database. This database is provided by Tsukuba

Fig. 1. Typical image from a camera with a hyperbolic mirror

Fig. 2. Panorama image derived from the previous image

Research Center, Japan, for the Real World Computing Partnership and contains a dataset of about 8000 robot positions and associated measurements from sonars, infrared sensors and omni-directional camera images. The measurements in the database were obtained by positioning the robot (a Nomad 200) on the grid-points of a virtual grid with distances between the grid-points of 10 cm. One of the properties of the Nomad 200 robot is that it moves around in its environment while the sensor head maintains a constant orientation. Because of this, the state of the robot is characterized by the position $\mathbf{x}$ only. The omni-directional imaging system consists of a vertically oriented standard color camera and a hyperbolic mirror mounted in front of the lens. This results in images as depicted in figure 1. Using the properties of the mirror we transformed the omni-directional images to 360 degrees panoramic images. To reduce the dimensionality we smoothed and subsampled the images to a resolution of 64×256 pixels (figure 2).

A set of 2000 images was randomly selected from the total set to derive the eigenimages and associated eigenvalues. We found that for 80% reconstruction error we needed about 90 eigenvectors. However, we are not interested in the reconstruction of images, but in the use of the low-dimensional representation for robot localization. In figure 4 the first 3 eigenimages (i.e. those with the largest corresponding eigenvalues) are displayed.

Fig. 3. The environment from which these images were taken.

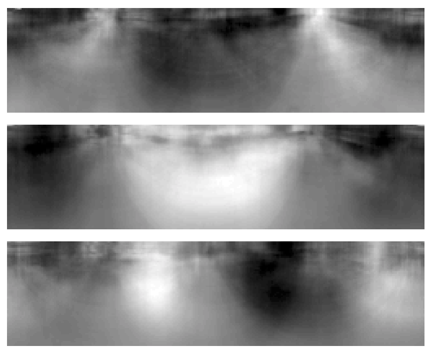


Fig. 4. The first 3 eigenvectors.

6.2 Observation model

In section 4 we described the kernel estimator as a way to represent the observation model $p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$. In such a method usually all training points are used for the modeling. In our database we have 8000 points which we can use. If we use this whole dataset this means that in the operational stage we should calculate the distance to all 8000 points for the localization, which, even though the dimensions of $\mathbf{x}$ and $\mathbf{y}$ are low, is computationally too slow. We are therefore interested in taking only a part of these points in the kernel density estimation model. In the following sections a set of about 300 images was selected as a training set. These images were taken from robot positions on the grid-points of a virtual grid with distances between the grid-points of 50 cm.

Another issue in the kernel method is a sensible choice for the width of the Gaussian kernel.The optimal size of the kernel depends on the real distribution (which we do not know), the number of kernels and the dimensionality of the problem. When modeling $p(\mathbf{x}, \mathbf{y})$ for a one-dimensional feature vector $\mathbf{y}$ with our training set we found that $h \approx 0.1$ maximized the likelihood of an independent test set. The test set consisted of 100 images, randomly selected from the images in the database not designated as training images. When using more features for localization (a higher dimensional feature vector $\mathbf{y}$) the optimal size of the kernel was found to be higher. In section 6.4 we study the effect of the kernel size on the localization model.

6.3 Localization

In a Markov localization procedure, an initial position estimate is updated by the features of a new observation using an observation model 4. The initial position estimate is computed using the motion model, and gives an informed prior in the Bayes rule. Since we are only interested in the performance of the observation model, we assume a flat prior distribution on the admissible positions $\mathbf{x}$.

In the current experiments we studied how many of the principal components are needed for good localization In figure 5 we see the distribution $P(\mathbf{x}|\mathbf{y})$ for an image which was taken at position (1.74, -0.96), for two different number of features: five eigenimages or one eigenimage.

a) b)

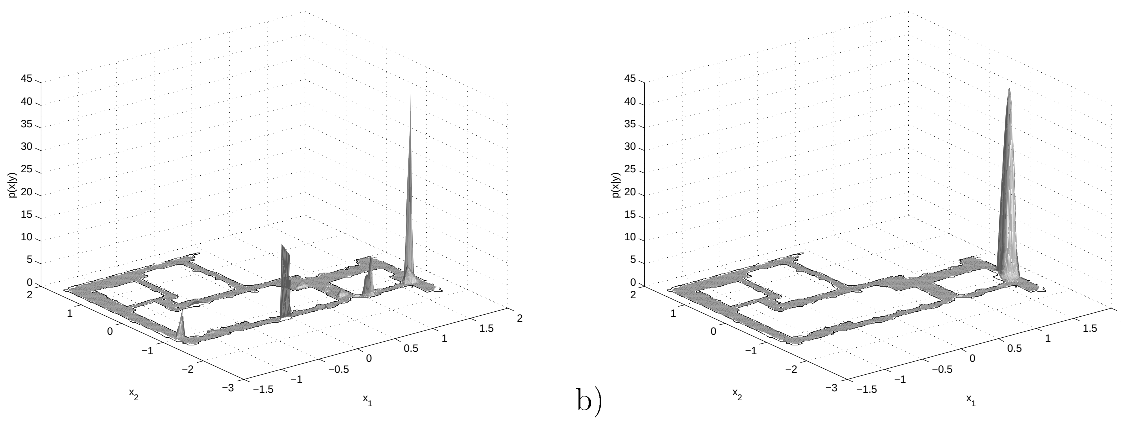
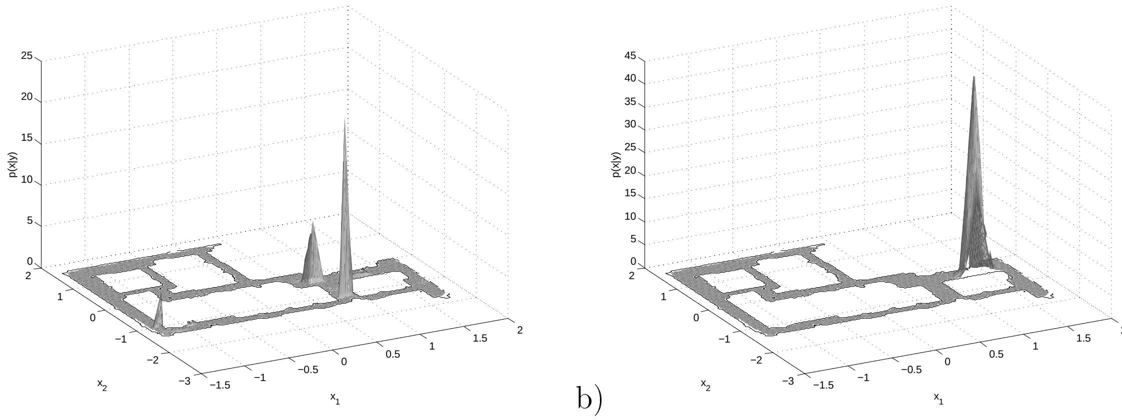


Fig. 5. An image at position (1.74, -0.96) is taken. The figure depicts the probability distribution over the learned locations. a) The first eigenvector (with the highest eigenvalue) is used as feature. b) The first 5 eigenvectors are used.

We observe that the distribution when using a single feature has multiple peaks, indicating multiple hypotheses for the position. This is solved if more features are taken into account. In both situations the maximum a posteriori value is close to the real robot position. This is not always the case. If we take another position (in this case (1.32,-0.37)), a wrong position estimate is obtained if too few features are used, see figure 6.



a) b)

Fig. 6. An image at position (1.32,-0.37) is taken. The figure depicts the probability distribution over the learned locations. a) Only the first component is used. b) 5 eigenvectors are used.

For our dataset usually less than 10 features are needed to have the maximum a-posteriori probability at the real robot position. However, in some exceptional cases we got an erroneous localization, even if more features are used. The reason for this is that appearances at two spatially different positions may be similar, an effect which is sometimes called 'perceptual aliasing'. Figure 7 gives an example what is happening. The image at the test position is shown on top, and the image reconstructed from the low-dimensional $(20-D)$ representation of this image is shown in the middle. On the basis of this $20-D$ feature vector the $P(\mathbf{x}|\mathbf{y})$ is computed, and the position $\mathbf{x}$ where the distribution is maximal. Now we looked at the learning image which is closest to this maximum, and this image is depicted at the bottom of Figure 7. In the low-dimensional representation this image is apparently closer to the test image than one of the learning images taken around the test image. Of course such erroneous localizations disappear if the motion model is used to give an informed prior if the full Markov localization is used.

In the following we will show how the performance of the observation model depends on the model parameters and on the number of features.

6.4 Effect of the kernel width of the Parzen estimator

In section 4 we described the kernel density estimation method which is used to represent the observation model. We want to investigate the effect of the kernel width to the localization performance. Localization was carried out with 1 resp. 2 eigenvectors. In figure 8 we see how both the loss-based risk (the average localization error) and the entropy-based risk (both normalized) depend on the width h of the kernel in the Parzen method.

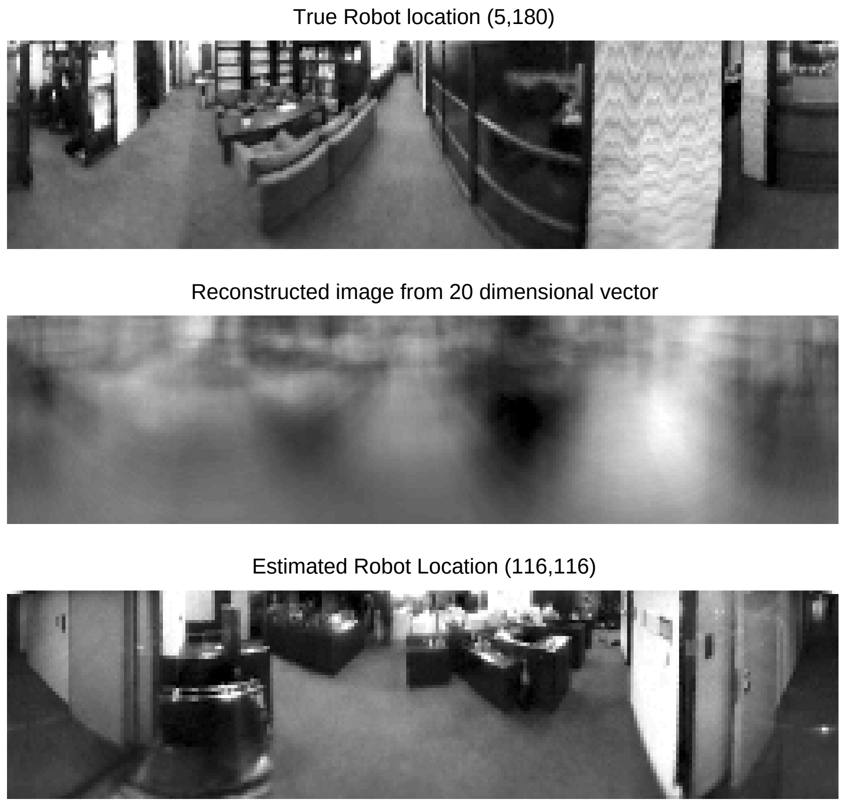


Fig. 7. a)An image at a test location, b) The representation in the feature space and c) View from the position with maximal a posteriori probability.

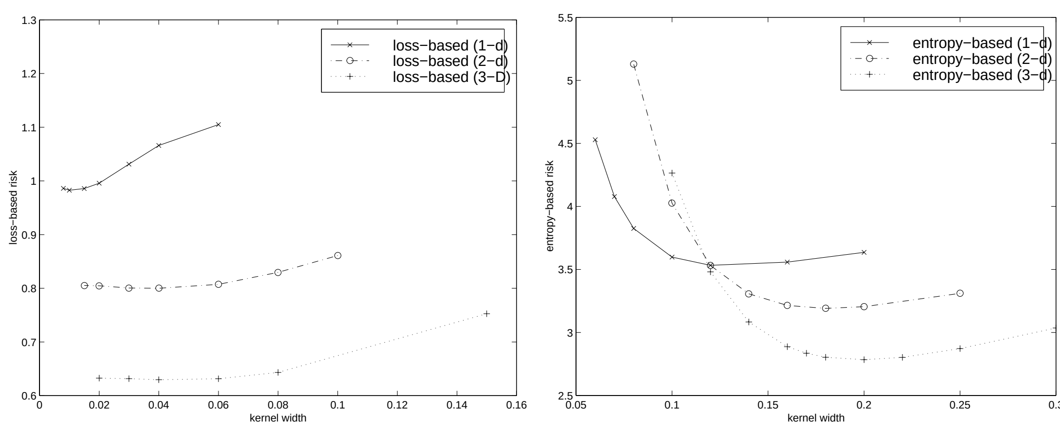


Fig. 8. Performance of the model for different kernel size.

Remarkable is the difference between the two performance criteria. The optimal risk according to the entropy criterion is achieved with a $h \approx 0.1$ for 1-d features and $h \approx 0.14$ for 2-d features, while the optimal width according to the loss criterion is much smaller. The reason for this is that the entropy based criterion uses the joint probability $p(\mathbf{x}, \mathbf{y})$ and thus requires smoothing in a higher dimensional space that the loss-based criterion, which uses only the conditional probability $p(\mathbf{y}|\mathbf{x})$.

As we discussed earlier, using more features generally give better results. Here we present the localization risk as a function of the number of features q. Note that in this experiment we always use the q features with the highest eigenvalues. In figure 9 we plotted the performance of the localization method for different number of features.

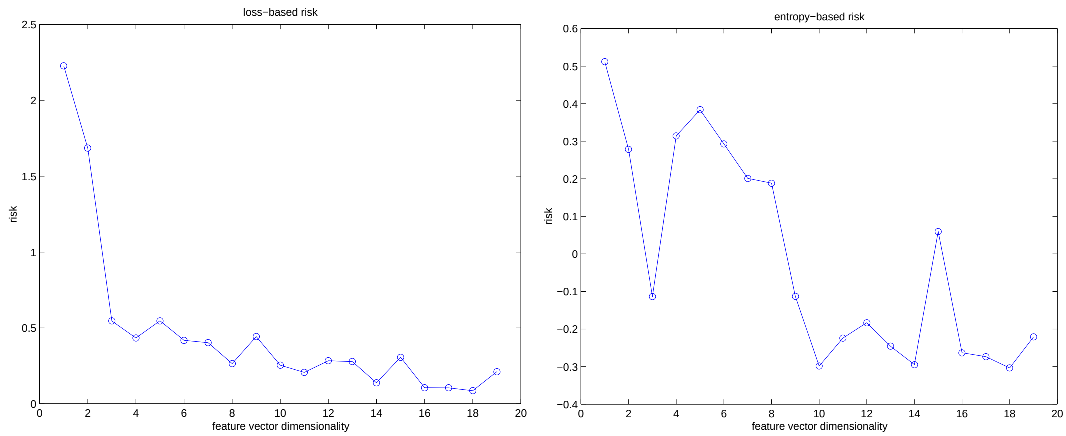


Fig. 9. Performance for different number features. (left) Loss-based risk. (right) Entropy-based risk.

In the previous experiments we always took the features with the highest eigenvalues. This can be argumented if we use the features for the *reconstruction* of the image, but is not necessarily so if we take the features for localization. To see whether features (eigenvectors) with highest eigenvalues are more important for localization than features with lower eigenvalues we looked at the performance for the individual features. We see that in this case the features

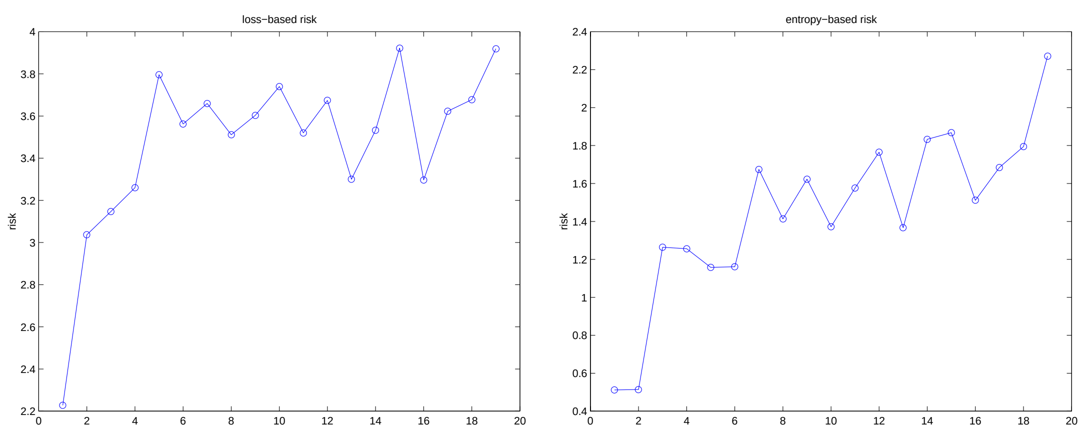


Fig. 10. Performance for different individual features. (left) Loss-based risk. (right) Entropy-based risk

with highest eigenvalues are indeed most important for the localization.

7 Discussion and conclusions

We showed that an appearance-based method gives good results on localizing a mobile robot in an office environment of about 17 x 17 m. The average expected localization error from our test set is about 40cm if around 15 features are used and the environment is represented with 300 training samples.

Note that we studied the worst-case scenario: the robot has no prior information about its position (the 'kidnapped robot' problem), and combined with a motion model the localization accuracy should be better. A second observation is that the environment can be represented by only a small number of parameters. For the 300 15-dimensional feature vectors the storage capacity is almost negligible and the look-up can be done very fast.

The experiments on the sensitivity of the method for the parameters of the kernel density estimation show that this model plays an important role in the localization. Care has to be taken in choosing the right value for the kernel width. Instead of taking a kernel method for approximating the distribution also other methods as for example neural networks can be used. In previous work we looked at some other methods [23]. However, also in those methods some free parameters have to be chosen.

The experiments were carried out with an extensive dataset, with which we were able to get good estimates on the accuracy of the method. However, the data were obtained in a static environment, with constant lighting conditions. Our current research in this line focuses on investigating which features are most important if also changes in the illumination will take place.

8 Acknowledgment

We would like to thank the people in the Real World Computing Partnership consortium and the Tsukuba Research Center in Japan for providing us with the MEMORABLE robot database.

References

[1] W. Burgard, A. Cremers, D. Fox, G. Lakemeyer, D. Hähnel, D. Schulz, W. Steiner, Walter, and S. Thrun. The interactive museum tour-guide robot. In A. P. Press, editor, *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 1998.

[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] K. Deguchi. A direct interpretation of dynamic images and camera motion for vision guided robotics. In *Proc. IEEE/SICE/RSJ Int. Conf. on Multisensor Fusion*, pages 313–320, 1996.

[4] K. Fukunaga. *Introduction to statistical pattern recognition.* Academic press, New York, 1972.

[5] M. Ishikawa, S. Kawashima, and N. Homma. Memory-based location estimation and navigation using bayesian estimation. In *Proc. of ICONIP'98*, pages 112–117, 1998.

[6] M. Jägersand. Image based view synthesis of articulated agents. In *Proc. of Comp. Vision and Pattern Recognition*, 1997.

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

[8] K. Konolige and K. Chou. Markov localization using correlation. In *Proc. International Joint Conference on Artificial Intelligence*, pages 1154–1159. Morgan Kauffmann, 1999.

[9] B. Kröse and R. Bunschoten. Probabilistic localization by appearance models and active vision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2255– 2260, 1999.

[10] 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.

[11] M. J. Matarić. Integration of representation into goal-driven behavior-based robots. *IEEE Trans. on Robotics and Automation*, 8(3):304–312, 1992.

[12] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *Int. Jrnl of Computer Vision*, 14:5–24, 1995.

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

[14] F. Pourraz and J. Crowley. Continuity properties of the appearance manifold for mobile robot estimation. In *Proc SIRS'98*, Edinburgh, 1998.

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

[16] Pierce, A and B. Kuipers: "Learning to explore and build maps". Proc. AAAI-94, AAAI/MIT Press 1994.

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

[18] M. Schmitt, M. Rous, A. Matsikis, and K. Kraiss. Vision-based self-localization of a mobile robot using a virtual environment. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, pages 2911–2916, Detroit, USA, 1999.

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

[20] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998.

[21] Nikos Vlassis and Ben Kröse. Mixture conditional density estimation with the em algorithm. In *Proc. ICANN'99, 9th Int. Conf. on Artificial Neural Networks*, pages 821-825, 1999.

[22] N. Vlassis, Y. Motomura, and B. Kröse. An information-theoretic localization criterion for robot map building. In *Proc. ACAI'99, Int. Conf. on Machine Learning and Applications*, Chania, Greece, July 1999.

[23] N. Vlassis and B. Kröse. Robot environment modeling via principal component regression. In *IROS'99, Proceedings of 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 677–682, 1999.