Enhancing Appearance-based Robot Localization Using Sparse Disparity Maps

Josep M. Porta, Jakob J. Verbeek and Ben Kröse
IAS Group, Informatics Institute, University of Amsterdam
Kruislaan 403, 1098SJ, Amsterdam, The Netherlands
{porta,jverbeek,krose}@science.uva.nl

***Abstract*— In this paper, we enhance appearance-based robot localization by using disparity maps. Disparity maps provide the same type of information as range based sensors (distance to objects) and thus, they are likely to be less sensitive to changes of illumination than plain images, that are the source of information generally used in appearance-based localization. The main drawback of disparity maps is that they can include very noisy depth values: points for which the algorithms can not determine reliable depth information. These noisy values have to be discarded resulting in missing values. The presence of missing values makes Principal Component Analysis (the standard method used to compress images in the appearance-based framework) unfeasible. We describe a novel Expectation-Maximization algorithm to determine the principal components of a data set including missing values and we apply it to disparity maps. The results we present show that disparity maps are a valid alternative to increase the robustness of appearance-based localization.**

I. INTRODUCTION

In appearance-based localization [5], the environment is not represented explicitly, but implicitly as a database of images (or features derived from them) collected at known positions in a training phase. For localization, the features of the image observed by the robot are compared with the features stored in the database.

A problem is that images largely change due to variations on illumination and so do the corresponding features. Consequently, illumination changes can largely degrade the performance of the appearance-based localization techniques. A possible solution to this problem is to complement intensity information with information provided by range sensors such as sonar or laser scanners.

In this paper, we describe a method which uses the depth information from a stereo vision system to enhance the appearance-based localization. Depth profiles, or *disparity maps*, are computed by matching points on two images. If the depth is only computed for few of the points in the images we obtain a *non-dense* disparity map that do not provide much more information that that provided by sonars or laser scanners. If we compute the depth for all pixels then we get a *dense* disparity map [1]. The advantage of this kind of maps is that, in contrast to standard laser range finders, which provide a 1-D depth map, a more informative 2-D depth map is determined. Additionally, dense disparity maps could be directly used for appearance-based localization as if they were plain images. The problem of dense disparity maps is that they can include very noisy depth values (especially in non-textured areas in the scene). If these non-reliable depth values are taken apart, we get a *sparse* disparity map, that is, a disparity maps that include some missing values. The issue of missing values particularly plays a role in the way we extract features for the appearance model. For the intensity images, we use features extracted from a Principal Component Analysis (PCA) on the set of collected images [5]. The numerically most accurate way to compute the features from the image set is by taking the Singular Value Decomposition (SVD) of the image matrix. However, because of the missing values, this technique can not be applied to sparse disparity maps.

In this paper, we propose the use of a Expectation-Maximization (EM) algorithm to determine the principal components for a set of disparity maps including missing values. This algorithm is similar to that described in [8], but we introduce a simpler way to deal with the missing values that makes the whole process more efficient. Thanks to this EM algorithm sparse disparity maps can be used in appearance-based localization.

This paper is organized as follows. First, we introduce the general framework for appearance-based localization. Next, we briefly describe how disparity maps are defined from intensity images. In Sections IV to VI, we introduce the EM algorithm to extract features from the disparity maps and we describe how these features can be used in our appearance-based localization framework in addition to features derived from intensity images. Next, we present results obtained in a real environment and, finally, we conclude summarizing our work and extracting some conclusions out of it.

II. APPEARANCE-BASED LOCALIZATION

Appearance-based localization departs from a set of training images $Z = (z_1, \ldots, z_N)$ taken at known poses (position/orientation) $X = (x_1, \ldots, x_N)$. Images are linearly compressed to get a set of features for each image $Y = (y_1, \ldots, y_N)$, where

$$y_i = Cz_i. \tag{1}$$

The projection matrix C expands the principal components of the data set Z and it is computed off-line applying PCA. The rows of C are the eigenvectors corresponding to the d largest eigenvalues.

In the on-line execution, we aim at estimating the probability on the robot's pose at time t, $p(x_t)$. This is usually done assuming that the environment is Markovian and deriving $p(x_t)$ from $p(x_{t-1})$, from a known *action model*, $p(x_t|u_{t-1},x_{t-1})$ (with u_{t-1} the action executed by the robot), and from a *sensor model*, $p(y_t|x_t)$. In the appearance-based framework, the sensor model is defined using the training set and the current observation, y_t. For instance, Vlassis *et al.* in [10] propose a nearest-neighbor formulation for the sensor model

$$p(y|x) = \sum_{j=1}^{J} \lambda_j \phi(x|x_j), \tag{2}$$

with x_j the J nearest neighbors (i.e., the subset of training points obtained at positions x_j and with an associated set of features y_j more similar to the features of the current observation y_t), λ_j a set of weights that favor closer nearest neighbors, and ϕ a Gaussian. The closer the training points used to define the sensor model to the actual position of the robot, the smaller the error in localization. Changes in illumination result in wrong matches between the features of the current image and those in the training set and, thus, in a wrong sensor model and in a wrong update of the robot's position. To minimize this problem as much as possible, we propose to make use of the information provided by disparity maps.

III. DISPARITY MAPS

Disparity maps are determined matching points in images taken by a pair of calibrated cameras mounted on the robot. Given a single image, the three-dimensional location of any visible object point Q must lie on the line that passes through the center of projection of the camera c and the image of the object point p. The determination of the intersection of two such lines generated from two independent images is called triangulation and provides the 3-D position of Q w.r.t the cameras.

To apply triangulation, for each pixel p in one of the images we have to search for a correspondent point in the other image. The epipolar constraint reduces this search to a single line called the epipolar line [3]. Usually, the correspondence is done by comparing areas around pixel p with areas around each candidate pixel p'. The most similar pixels p and p' are assumed to correspond to different projections of the same point Q in the scene. If the images planes for the two cameras are co-planar, the distance r from the scene point Q to the cameras can be computed as

$$r = \frac{b\,f}{d-d'},$$

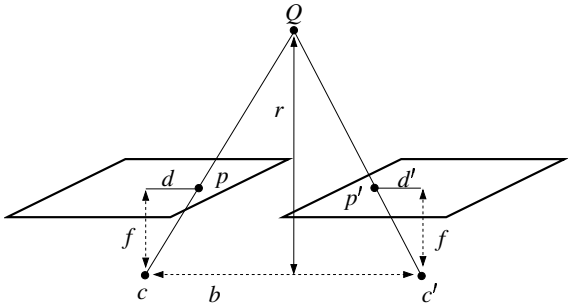

Fig. 1. Elements in disparity computation.

where b is the baseline (distance between the two viewpoints), f is the focal length of the cameras, d is the horizontal distance from the projected point p to the center of one of the images, and d' is the same for the other image (see Figure 1). The difference $d-d'$ is called disparity. With constant baseline b and focal length f, the disparity is inversely proportional to the depth of point Q.

The stereo algorithm we use [4] applies many filters in the process to determine the disparity map both to speed up the process and to ensure the quality of the results. For instance, if the area around pixel p is not textured enough it would be very difficult to find a single corresponding point p' (we are more likely to end up with many points p' with almost the same probability of being the corresponding point of p). For this reason, pixels on low textured areas are not even considered in the matching process. The result of this and other filtering processes to avoid too noisy disparity values is to produce a sparse disparity map: a disparity map where many pixels do not have a disparity value (see Figure 2). This makes the use of standard PCA to determine the projection matrix unfeasible and we have to use more elaborated techniques as the EM algorithm introduced in the following sections.

IV. THE EM ALGORITHM

The EM algorithm [2], [6] aims at determining the maximum likelihood model for a given set of data Z. Not all possible models are considered, but only those inside a family of models defined by a set of parameters, collectively denoted as θ (in our application, these parameters are a projection matrix C and a parameter σ that indicates the noise in the data out of the principal component subspace). The optimal model is determined by maximizing the likelihood or, equivalently, the log-likelihood:[1]

$$\Phi(\theta) = \log p(Z;\theta).$$

In many cases, Φ can be defined using an auxiliary set of unobserved or *hidden* variables h. In this case, we have

$$\Phi(\theta) = \log \int_h p(Z,h;\theta).$$

[1]The log is used since it makes the resulting expressions simpler.

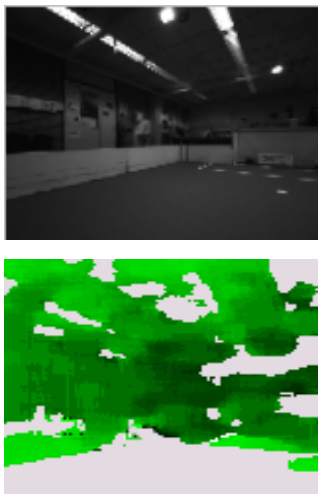

Fig. 2. Plain image (top) and the corresponding disparity map (bottom). In the disparity map, light gray areas are missing values.

Φ might be maximized using a gradient ascent process. However, the EM algorithm provides a simpler to implement and step-size free algorithm using iterative lower-bound maximization.

For a given set of parameters θ, the EM algorithm first finds a lower bound Ψ of Φ, possibly such that Ψ touches Φ at point θ. After that, the lower bound Ψ is maximized for the parameters θ. The definition of Ψ for a fixed set of parameters θ is the E step of the algorithm and the maximization of the resulting Ψ is called the M step. The sequential iteration of E and M steps from an arbitrary initial set of parameters θ is guaranteed to find a (local) maximum of Φ.

The lower-bound Ψ used in the E step, can be defined as

$$\Psi(\theta) = \Phi(\theta) - KL(q(h)\|p(h|Z;\theta)) \leq \Phi(\theta),$$

where KL denotes the (non-negative) Kullback-Leibler divergence between two probability distributions. To make $\Psi(\theta) = \Phi(\theta)$ for the current parameters θ, the KL should be zero. The KL divergence is zero iff the two compared probability distributions are equal. Thus, in the E step we set $q(h) = p(h|Z;\theta)$.

In the M step, we have to maximize Ψ. We can rewrite Ψ as

$$\begin{aligned}\Psi(\theta) &= \Phi(\theta) - \int q(h)\frac{q(h)}{p(h|Z;\theta)} = \\ &= -E_{q(h)}\left[\log\frac{q(h)}{p(h|Z;\theta)}\right] + \log p(Z;\theta) = \\ &= E_{q(h)}\left[\log\frac{p(Z,h;\theta)}{q(h)}\right] = \\ &= \int_h q(h)\,\log p(Z,h;\theta) - q(h)\,\log q(h).\end{aligned}$$

The relevant term of Ψ to be maximized is

$$\Psi_M = \int_h q(h)\,\log p(Z,h;\theta)$$

since the other term of Ψ does not depend on θ.

For a more detailed presentation of the EM algorithm see [6], which we followed in the above brief description.

V. APPLYING EM TO DISPARITY MAPS

Given a $D \times N$ matrix Z, representing a set of disparity maps $Z = \{z_1, \ldots, z_N\}$, our aim is to find a projection that maps Z onto its d-dimensional principal components space ($d \ll D$). For each disparity map z_i, we have a subset of elements that are observed $z_{i,o}$ and a set of missing values $z_{i,h}$. Thus, we want to define the probability

$$p(y_i|z_{i,o}),$$

with y_i the set of principal component features derived from the observed values in disparity map z_i. We do so by first defining $p(y)$ as a Gaussian with zero mean and identity covariance, and $p(z|y) = \mathcal{N}(z;C^\top y, \sigma^2 I)$. The projection density $p(y_i|z_{i,o})$ then follows a Gaussian distribution $\mathcal{N}(y_i;\mu_{y_i},\Sigma_{y_i})$, where $\Sigma_{y_i}^{-1} = I + C_o C_o^\top/\sigma^2$ and $\mu_{y_i} = \Sigma_{y_i} C_o z_o/\sigma^2$. Here, C is the $d \times D$ projection matrix from the space of disparity maps to the space of principal components, $C^\top$ performs the inverse mapping, and C_o contains only the columns of C that correspond the observed elements in a given map z_i. Observe that C_o is different for each disparity map and hence so is the covariance matrix Σ_{y_i}. As mentioned, the parameters that describe our model are the projection matrix C and σ.

To find the maximum likelihood projection matrix C, we have to particularize the general E and M steps described in the previous section. This can be done paying attention only to the observed values z_o, taking the projected features as the hidden variables, and not making inference about the missing pixels. However, the EM algorithm can be made more efficient by estimating y and z_h at the same time.

In the E step the KL divergence to be minimized is

$$KL = \int q(y,z_h)\frac{q(y,z_h)}{p(z_h,y|z_o;C,\sigma)}.$$

In order to obtain an efficient E step we factor q over y and z_h. Thus,

$$\begin{aligned}KL = \int q(y)\Bigg[&\log\frac{q(y)}{p(y|z_o;C,\sigma)} \\ &+ \int q(z_h)\log\frac{q(z_h)}{p(z_h|y;C,\sigma)}\Bigg].\end{aligned}$$

Using free-form optimization, we have

$$q(z_h) \propto \exp\int q(y)\log p(z_h|y;C,\sigma),$$

$$q(y) \propto p(y|z_o;C,\sigma)\, \exp \int q(z_h) \log p(z_h|y;C,\sigma).$$

Both of these densities are Gaussian. The covariance of the $q(z_h)$ is given by $\sigma^2 I$ and the mean for the missing values on the disparity map i is given by $z_{i,h} = C_h^\top y_i$. The covariance and the means (collected as columns in Y) of the $q(y)$ for the disparity maps are given respectively by

$$\Sigma_y = [I + \sigma^{-2} C C^\top]^{-1},$$
$$Y = \sigma^{-2} \Sigma_y C Z,$$

where Z is the data matrix with missing values filled in with the new $z_{i,h}$.

Observe that, the covariance matrix Σ_y is common for all data points when computing the set of projections Y. Therefore, we only have to perform one matrix inversion and not one per data point.

With respect to the M step, the relevant part of the likelihood function to be optimized is

$$\Psi_M = \int_{y,z_h} q(y,z_h)\, \log p(z_o,z_h,y;C,\sigma).$$

Analytical manipulation leads to the effective objective to be maximized:

$$\begin{aligned}\Psi_M =& ND \log \sigma^2 + \\ &+ \sigma^{-2} \left[\sum_{i=1}^{N} \|z_i - C^\top y_i\|^2 + Tr\{C^\top \Sigma_y C\} \right] + \\ &+ \sigma^{-2} D_h \sigma_{\text{old}}^2,\end{aligned}$$

with D_h the total amount of missing values in Z and σ_{old} the previous value for σ. From the above, we get the simple updates

$$C = ZY^\top \left(N\Sigma_y + YY^\top \right)^{-1}$$
$$\sigma^2 = \frac{1}{ND} \left[NTr\{C^\top \Sigma_y C\} + \sum_{i=1}^{N} \|z_i - C^\top y_i\|^2 + D_h \sigma_{\text{old}}^2 \right].$$

The E and M steps have to be iterated while there is a large (relative) change in the lower bound of the log-likelihood function that, after the updates, reads to

$$\begin{aligned}\Psi(C,\sigma) = &-\frac{N}{2} \left(D \log \sigma^2 + \text{Tr}\{\Sigma_y\} - \log |\Sigma_y| \right) \\ &- \frac{1}{2} \text{Tr}\{YY^\top\} + \frac{1}{2} D_h \log \sigma_{\text{old}}^2.\end{aligned}$$

A reasonable initialization of C would be that containing d randomly selected disparity maps (with zeros in the missing values) and σ^2 equal to the initial reconstruction error (for the observed values).

Each iteration of the EM steps scales with order $O(dDN)$, assuming $d < N$. Our approach to finding the maximum likelihood C and σ has the advantage that the guess of the missing values comes essentially *for free* while in other approaches [8] they were obtained extending the E step with a optimization process in the space of images for every image separately. This is an important saving especially when working with data sets such as space disparity map where each point z contains many missing values.

Once C and σ are determined, the set of features y corresponding to a new disparity map z are computed as

$$y = (\sigma^2 I + C_o C_o^\top)^{-1} C_o z_o.$$

The matrix $(\sigma^2 I + C_o C_o^\top)^{-1} C_o$ maps from (observed) disparity values to feature detectors and plays the same role as matrix C in equation 1. For vanishing sigma, $(\sigma^2 I + C_o C_o^\top)^{-1} C_o$ is the pseudo inverse of C_o.

The most expensive step in the computation of the vector of feature detectors for each disparity map is the product $C_o C_o^\top$ that is $O(Dd^2)$. Because of this, the on-line determination of the features is only feasible for relative small d (and moderated D's).

A Matlab version of the EM algorithm just described can be downloaded from [9].

The just described algorithm can be applied to any set of data with missing values. Thus, we can apply it to disparity maps but we can also apply it to a concatenation of the intensity image and the disparity map obtained at a given position. The advantage of concatenating intensity and disparity maps is that the information from the intensity images can eventually help to deal with missing values in the disparity maps. The problem of proceeding this way is that changes in illumination would largely modify part of the input vector (that relative to intensity) and, as a consequence, we would lose the robustness to changes in illumination offered by disparity maps.

VI. SENSOR MODEL FUSION

Once we have a way to define features from disparity maps, it remains the question of how to combine the information coming from disparity with that obtained from intensity to define a unified sensor model. Two possible solutions come to mind: to combine them in a conjunctive way or in a disjunctive one.

A conjunctive-like combination can be achieved factorizing the sensor model

$$p(y_d, y_i|x) = p(y_d|x)\, p(y_i|x),$$

with y_d the features obtained for disparity and y_i those for intensity. In this way, only those training points consistent with both the current intensity image and the current disparity map are taken into account to update the robot's position. The problem of this formulation is that wrong matches for intensity (or for disparity) would result in an almost null sensor model and, thus, the position of the robot would be updated almost without sensory information.

Fig. 3. Our robot with the stereo head used to take images and disparity maps.

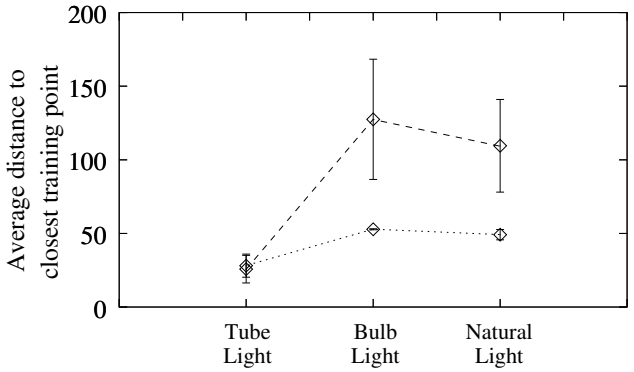

Fig. 4. Error in positioning in three different illumination conditions, using only intensity images (dashed line) and intensity and disparity images (dotted line).

To avoid this, we propose to use a disjunctive-like model that can be implemented defining the global sensor model as linear combination of the intensity and disparity sensors models

$$p(y_d, y_i|x) = w_d\, p(y_d|x) + w_i\, p(y_i|x),$$

that, using equation 2, reads to

$$p(y_d, y_i|x) = w_d \sum_{j=1}^{J} \lambda_j\, \phi(x|x_j) + w_i \sum_{j=1}^{J'} \lambda'_j\, \phi(x|x'_j),$$

where x_j and x'_j are the closest training points to the features of the current disparity and intensity images respectively. The weights w_d and w_i (with $0 \leq w_d \leq 1$, $w_i = 1 - w_d$) can be used to balance the importance of the information obtained with each type of sensor. If both information sources are assumed to be equally reliable, we can set $w_d = w_i = 1/2$.

With this expression for the sensor model, all the likely hypotheses on the robot position suggested by the current observation are taken into account. The particle filter we use to update the probability on the robot's position [7], [10] takes care of filtering the noise and, thus, of preserving the hypothesis that is more consistent over time.

VII. EXPERIMENTS AND RESULTS

To assess the contribution of using disparity maps in appearance-based localization, we collected a training set moving our robot (see Figure 3) in an area of 900×500 centimeters. Training points were disposed on a regular grip with a resolution of 50 cm. At each point on the grid, we took intensity images and disparity maps rotating the camera every 10°. This makes a total amount of more than 4500 training images. Each image (and each disparity map) has 160×120 pixels. Therefore, we have $D = 19200$ pixels per image. In this training set, about 20% of the pixels on the disparity maps are undefined.

We use $N = 100$ randomly sampled images to compute the $d = 20$ principal components of the training set. For intensity images, we used a standard PCA technique to determine the principal components and, for disparity maps, we applied the algorithm we describe in Section V.

The principal component of the disparity maps are computed in 13 iterations of our EM-based algorithm with a convergence threshold (relative change on two consecutive maximums of functions Ψ) of 0.001. The execution of this process takes 125 seconds in a Pentium 4 at 2.0 GHz.

To test the our localization system, we moved the robot along a pre-defined path in three different illumination conditions: using tube lights (that was the light used to get the training set), using bulb lights and, finally, with natural light (obtained opening the curtains of the windows all along one of the sides of the lab). These three different illuminations ensure not only a difference in the global intensity of the light, but also a change in the distribution of light sources. At regular distances along the test path, we take an image and we compute the corresponding sensor model using the J training points with a set of features more similar to those corresponding to the just obtained image. As mentioned before, the closer the training points used to define the sensor model to the actual position of the robot, the better the sensor model and, thus, better the update of the robot position estimation.

Figure 4 shows the average (and the variance) of the error in the sensor model

$$e = \min_{\forall n} \|r - n\|,$$

with $r = (r_x, r_y, r_\phi)$ the pose of the robot at the test position and $n = (n_x, n_y, n_\phi)$ the pose of the points used to define

the sensor model (that are different for each test position). An error in the range $[25, 50]$ is quite reasonable since the distance between training points in X and Y dimensions is 50 cm.

We repeat the test in two cases: (a) using only intensity images (dashed line on Figure 4) and (b) using, additionally, disparity maps (dotted line on the figure). In the first case we use $J = 10$ (quite small compared with the total number of images in the training set) and in the second case we use $J = 5$ but for both intensity and disparity (so, we also get 10 matching points).

We can see that the use of disparity maps results in a reduction of the error when illumination is different from that in which the training set was obtained. Additionally, the use of disparity result in a reduction of the variance of the error. This means that the error when using disparity is smaller in (almost) all test points (and not only in few of them).

The computation of the features for a given disparity image takes less than 0.1 seconds (also in a Pentium 4 at 2.0 GHz). Therefore, the use of disparity maps for appearance-localization is completely feasible.

VIII. CONCLUSIONS

In this paper, we have introduced the use of sparse disparity maps to increase the robustness of appearance-based localization to changes in illumination.

To compress sparse disparity maps to a reduced set of features, we have to deal with the problem of missing values. We have presented a novel EM-based algorithm to extract the principal components of a set of data that is more efficient in the way in which it deals with missing values than previously existing methods.

After the dimensionality reduction, it remains the open problem of how to merge the information provided by disparity maps with that provided by intensity images. We have proposed a simple linear combination of the sensors models build separately for each type of sensor. In this way, noise in one sensors do not corrupt the information provided by the other sensor.

We have shown that, using features from disparity maps in addition to those obtained from intensity images, we can improve the quality of the sensor model when illumination conditions are different from those in which the training set is obtained. Thus, disparity maps are a good option to increase the robustness of appearance-based robot localization.

In the EM algorithm we have introduced, the value for missing pixels for a given disparity map are inferred using information from other disparity maps. In our future work, we would like to explore the possibility doing this using also information provided by the intensity images (close points with similar intensity value are likely to have similar disparity).

The main assumption behind our approach is the existence of a training set obtained off-line and densely sampled over the space where the robot is expected to move. In general, obtaining this training set is not a problem, but it would be desirable the robot to build it on-line. With this extension, our system would become a Concurrent Localization and Mapping (CLM) system. To achieve this improvement, we have to explore the use of incremental techniques for compressing the images that are obtained on-line.

IX. ACKNOWLEDGMENTS

This work has been partially supported by the European (ITEA) project *"Ambience: Context Aware Environments for Ambient Services"*.

We would like to thank Bas Terwijn for helping us to obtain the data necessary to perform the experiments reported in this paper.

X. REFERENCES

[1] L. Alvarez, R. Deriche, J. Sanchez and J. Weickert, "Dense Disparity Map Estimation Respecting Image Discontinuities: a PDE and Scale-Space Based Approach", *Technical Report RR-3874*, INRIA, 2000.

[2] A.P. Dempster, N.M. Laird and D.B. Rubin, "Maximum-Likelihood for Incomplete Data via the EM Algorithm", *J. Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[3] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, 1993.

[4] K. Konolige, "Small Vision System: Hardware and Implementation", *Eighth International Symposium on Robotics Research*, Japan, October, 1997.

[5] B.J.A. Kröse, N. Vlassis, R. Bunschoten and Y. Motomura, "A Probabilistic Model for Appearance-based Robot Localization", *Image and Vision Computing*, 19(6), 381–391, April, 2001.

[6] T. Minka, "Expected-Maximization as Lower Bound Maximization", `http://www.stat.cmu.edu/~minka/papers/em.html`.

[7] M.K. Pitt and N. Shephard, "Filtering via Simulation: Auxiliary Particle Filters", *J. Amer. Statist. Assoc.*, 94(446):590–599, June 1999.

[8] S. Roweis, "EM Algorithms for PCA and SPCA", in *Advances in Neural Information Processing Systems*, Vol. 10, The MIT Press, 1998.

[9] J.J. Verbeek, "Probabilistic PCA with Missing Values. Matlab Source Code", `http://www.science.uva.nl/~jverbeek/software`.

[10] N. Vlassis, B. Terwijn and B.J.A. Kröse, "Auxiliary Particle Filter Robot Localization from High-Dimensional Sensor Observations", *Proceedings of the IEEE International Conference on Robotics and Automations*, pages 7–12, Washington D.C., 2002.