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

Computing the Sensory Uncertainty Field of a Vision-based Localization Sensor

Amit Adam
Dept. of Mathematics

Ehud Rivlin
Dept. of Computer Science

Ilan Shimshoni
Dept. of Industrial Engineering

Technion – Israel Institute of Technology
Haifa 32000 – Israel
email: {amita,ehudr,ilan}@cs.technion.ac.il

Abstract

Recently it has been recognized that robust motion planners should take into account the varying performance of localization sensors across the configuration space. Although a number of works have shown the benefits of using such a performance map, the work on actual computation of such a performance map has been limited and has addressed mostly range sensors. Since vision is an important sensor for localization, it is important to have performance maps of vision sensors. In this paper we compute the performance map of a vision-based sensor. We show that the computed map accurately describes the actual performance of the sensor, both on synthetic and real images. The method we present (based on [6]) involves evaluating closed form formulas and hence is very fast. Using the performance map computed by this method for motion planning and for devising sensing strategies will contribute to more robust navigation algorithms.

1 Introduction

External sensors such as video cameras, laser range finders and sonar are being routinely used for mobile robot localization. Recently [10, 5, 7, 8, 11, 9] it has been recognized that the accuracy of the localization obtained by invoking the sensor, will in general depend on the configuration the robot is in. In other words, the combination of sensor and environment defines some kind of map which describes the quality of localization obtained at each configuration by using the sensor. We will refer to this map as the Sensory Uncertainty Field (SUF) which was coined in [10].

Existence of the SUF has led to some interesting higher level problems. A natural idea is to have the motion planner use the information in the SUF to plan routes which will pass in regions where the sensor works well, i.e. the sensory uncertainty is low. Such works may be found in [10], [5],[7]. In [8] a related notion is the information content of the environment at each configuration. Another similar idea motivated by visual servoing is described in [9]. Another use of the SUF map is demonstrated in [11], where the question of choosing the proper landmarks for localization is addressed. Celinski and McCarragher [3, 4] addressed the problem of sensing management - for example choosing the appropriate sensor to be used at each configuration. An SUF type of map is some of the input required for addressing such issues.

Although these higher level works illustrate the utility and importance of using and having an SUF, there has been limited work on actual computation of the SUF. The above mentioned works [10],[8],[7] all used a range sensor. The basic method of computing the SUF is to actually simulate the sensing algorithm at each configuration. The output of the algorithm on noisy measurements results in dispersed answers, and the SUF is some measure of this dispersion. Such a method for computing the SUF is time consuming because the simulation has to be run for each configuration in the configuration space. Therefore, various simplifications had to be employed in [10],[8].

In this work we use a fast method to compute a SUF for a vision sensor. This method does not involve simulation of the sensing. Instead we use a closed form formula which gives a direct estimate of the sensing output covariance matrix. Thus the method is much faster. To test this method, we compare the predicted covariance matrix with the actual scattering of results obtained by invoking the sensor many times (i.e. simulation), and show that the predicted covariance matrix indeed describes well this dispersion. This is shown both on synthetic and real images.

In the next section we describe the localization algorithm based on input from a vision sensor. Afterwards we describe the general method to estimate the covari-

0-7803-5886-4/00/$10.00© 2000 IEEE

ance matrix of the sensor output, and apply it to our specific case. Section 4 describes how we statistically tested the validity of our predicted covariance matrix. In section 5 we present comparison of the predicted vs. actual covariance matrices obtained in different scenes and configurations. The last section summarizes and concludes this paper.

2 Localization Algorithm

For localization we use a vision based sensor. We assume a camera is mounted on the robot. Assume a base image was taken from a known configuration. A second image is taken from the current configuration and a third image is taken after a small translation from the current configuration. By computing the camera egomotion between these three images we may deduce the current configuration. More details may be found in [2].

Our goal now is to predict the covariance matrix of the motion estimate we obtain. In this work we focused on the estimate of direction of translation, or focus of expansion (FOE).

We will now describe how we estimate the FOE using point matches between two images related by pure translation. Let $(p_i, p'_i) = (x_i, y_i, x'_i, y'_i)$ be the noise free (or pure) matches between the two views. Let $(\hat{p}_i, \hat{p}'_i) = (\hat{x}_i, \hat{y}_i, \hat{x}'_i, \hat{y}'_i)$ be the measured correspondences. Let us assume the error in each coordinate is a Gaussian random variable with zero mean and σ^2 variance. Let $S = \{(p_i, p'_i)\}$ be called the pure scene and let $\hat{S} = \{(\hat{p}_i, \hat{p}'_i)\}$ be called the measured scene.

Denote by $\mathcal{L}(S|\hat{S})$ the likelihood of the scene S being the true correspondences, given the measured scene $\hat{S}$. By the Gaussian assumption on the measurement noise, and by assuming independence between pixels and views, we obtain

$$\begin{aligned}\mathcal{L}(S|\hat{S}) &= \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x_1-\hat{x}_1)^2}{2\sigma^2}}\cdots\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(y_p-\hat{y}'_p)^2}{2\sigma^2}} \\ &= \frac{1}{(2\pi)^{2p}\sigma^{4p}}e^{-\frac{1}{2\sigma^2}[(x_1-\hat{x}_1)^2+\cdots+(y_p-\hat{y}'_p)^2]} \quad (1)\end{aligned}$$

Thus we think of the measures scene $\hat{S}$ as being a noise corrupted version of some true scene S, and the likelihood for each S is given by equation (1).

Assume that a certain point (u,v) is the FOE. Then it is well known that the segments $s_i = (p_i, p'_i)$ in the pure scene all lie on lines meeting at the point (u,v). Thus we are led to the following definition:

Definition: The scene $S = \{(p_i, p'_i)\}$ *supports* the point (u,v) if the lines through the pairs (p_i, p'_i) all meet at the point (u,v).

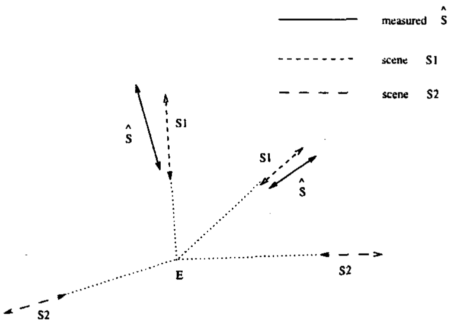

Figure 1: Possible supporting scenes for the point E

It is now natural to define the likelihood of the point (u,v) being the FOE, given the measured scene $\hat{S}$ as

$$\mathcal{L}(u,v) = \int_{\{S|S \text{ supports } (u,v)\}} \mathcal{L}(S|\hat{S}) \quad (2)$$

This definition is illustrated in figure 1. S_1 and S_2 are two possible scenes which support the fact that the point E is the FOE. Clearly given the measured scene $\hat{S}$, scene S_1 is much more likely.

It is clear that an infinite number of scenes may support the fact that a point (u,v) is the FOE. However, given a specific measured scene, the majority of these possible supporting scenes are very unlikely. In other words, the integrand in (2) nearly vanishes for the majority of the supporting scenes. Still, obtaining a good approximation of (2) may not be trivial. (See [12] for related work in which actual computation of simpler, one dimensional integrals resembling the integral in equation (2), was carried out in the context of curve fitting).

We chose to approximate (2) by taking only the largest integrand into account. For a point (u,v) we find the scene $\tilde{S}$ that supports (u,v) and that is also the most likely scene given $\hat{S}$ (with respect to all other scene supporting (u,v)):

$$\tilde{S} = \arg\max_{\{S|S \text{ supports } (u,v)\}} \mathcal{L}(S|\hat{S}) \quad (3)$$

and then we approximate (2) by

$$q(u,v) = \mathcal{L}(\tilde{S}|\hat{S}) \quad (4)$$

Finding the point (u,v) which maximizes the function q is our method for estimating the FOE, and we call this the *approximate maximum likelihood (AML)* estimate.

It remains to show how we can compute the scene $\tilde{S}$ that is the most likely scene which supports (u,v), given $\hat{S}$. Let us look at a specific segment $(\hat{p}_i, \hat{p}'_i)$ from

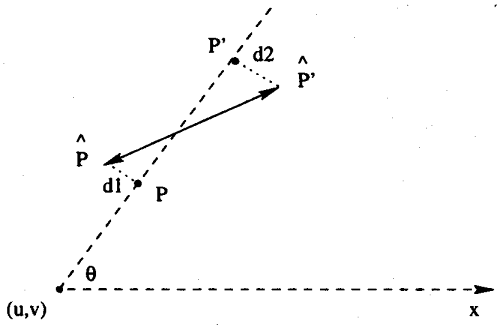

Figure 2: Definition of the function q

$\hat{S}$. We would like to move the points $\hat{p}_i$ and $\hat{p}'_i$ as little as possible to points p_i and p'_i, such that $(u,v), p_i$ and p'_i will all be colinear. The less we move $\hat{p}_i$ and $\hat{p}'_i$, the higher the likelihood of the new segment (p_i, p'_i). This is illustrated in figure 2. Thus, given the points $\hat{p} = (\hat{x}, \hat{y})$ and $\hat{p}' = (\hat{x}', \hat{y}')$, the geometric solution is as follows: pass a line through the point (u,v) such that the sum $d_1^2 + d_2^2$ of the distances from $\hat{p}$ and $\hat{p}'$ to the line, will be minimal. By simple calculus and geometry it may be verified that the line we are seeking creates an angle θ with the x axis such that

$$\theta = \frac{1}{2} \arctan \frac{2((\hat{x} - u)(\hat{y} - v) + (\hat{x}' - u)(\hat{y}' - v))}{(\hat{x} - u)^2 + (\hat{x}' - u)^2 - (\hat{y} - v)^2 - (\hat{y}' - v)^2} \tag{5}$$

where the arctangent is chosen in the range $[0, 2\pi)$ according to the signs of the numerator and denominator (as in the atan2 function).

Since the segments of the scene are independent, we may find for each measured segment the most likely (u,v)-supporting segment. The collection of these most-likely segments is the required scene $\tilde{S}$. The value of $q(u,v)$ is the sum of squared distances from the measured pixels to the line segments in $\tilde{S}$ (actually this is the logarithm of q up to a scale factor).

Once we have computed $\theta = \theta(u, v, \hat{x}, \hat{y}, \hat{x}', \hat{y}')$ the distances d_1, d_2 to the line are

$$\begin{aligned} d_1^2 &= (-\sin\theta(\hat{x} - u) + \cos\theta(\hat{y} - v))^2 \\ d_2^2 &= (-\sin\theta(\hat{x}' - u) + \cos\theta(\hat{y}' - v))^2 \end{aligned} \tag{6}$$

Since all segments are independent of each other, in order to find the AML FOE estimate we have to minimize w.r.t. (u,v) the function

$$\begin{aligned} F(u, v, \hat{x}_1, \ldots, \hat{y}'_p) &= \\ \sum_{i=1}^{p} \{(-\sin\theta(\hat{x}_i - u) + \cos\theta(\hat{y}_i - v))^2 &+ \\ (-\sin\theta(\hat{x}'_i - u) + \cos\theta(\hat{y}'_i - v))^2\} & \end{aligned} \tag{7}$$

3 Covariance Estimation

We now want to estimate the dispersion of the estimated $\widehat{(u,v)}$ at a certain configuration. Our estimate is based on a first order approximation derived in [6]. The input from the point matching algorithm is the set of correspondences $\hat{s}_i = (\hat{x}_i, \hat{y}_i, \hat{x}'_i, \hat{y}'_i)$ which are the noise corrupted versions of $s_i = (x_i, y_i, x'_i, y'_i)$. Denote by $X = (x_1, y_1, x'_1, y'_1, \ldots, y'_p)^t$ and $\hat{X} = (\hat{x}_1, \hat{y}_1, \hat{x}'_1, \hat{y}'_1, \ldots, \hat{y}'_p)^t$ the pure and noisy sets of matches. Let $\Theta = (u,v)$ be the correct FOE and $\hat{\Theta} = \widehat{(u,v)}$ be the estimated FOE. Then we have:

$$\begin{aligned} \Theta &= \arg\min_{\Theta} F(\Theta, X) \\ \hat{\Theta} &= \arg\min\nolimits_{\Theta} F(\Theta, \hat{X}) \end{aligned} \tag{8}$$

Let

$$g(\Theta, X) = \nabla_{\Theta} F = \frac{\partial F}{\partial \Theta}$$

Since $\hat{\Theta}$ minimizes $F(\Theta, \hat{X})$ and Θ minimizes $F(\Theta, X)$, we have that both $g(\hat{\Theta}, \hat{X})$ and $g(\Theta, X)$ are equal to 0. By using first order Taylor expansion, in [6] it is shown that this leads to

$$\frac{\partial g}{\partial \Theta}(\hat{\Theta}, \hat{X})(\hat{\Theta} - \Theta) = -\frac{\partial g}{\partial X}(\hat{\Theta}, \hat{X})(\hat{X} - X)$$

which leads to the following approximation of the covariance matrix of $\hat{\Theta}$:

$$\Sigma_{\hat{\Theta}} = M \Sigma_{\hat{X}} M^t \tag{9}$$

where

$$\begin{aligned} \Sigma_{\hat{X}} &= E[(\hat{X} - X)(\hat{X} - X)^t] \\ M &= (\frac{\partial g}{\partial \Theta})^{-1}(\frac{\partial g}{\partial X}) \end{aligned}$$

In our case we have

$$\begin{aligned} g &= \begin{pmatrix} F_u \\ F_v \end{pmatrix} \\ \frac{\partial g}{\partial \Theta} &= \begin{pmatrix} F_{uu} & F_{uv} \\ F_{vu} & F_{vv} \end{pmatrix} \\ \frac{\partial g}{\partial X} &= \begin{pmatrix} F_{u\hat{x}_1} & F_{u\hat{y}_1} & \cdots & F_{u\hat{y}'_p} \\ F_{v\hat{x}_1} & F_{v\hat{y}_1} & \cdots & F_{v\hat{y}'_p} \end{pmatrix} \end{aligned}$$

The matrix M is evaluated at the estimated $\hat{\Theta}$ and the measured $\hat{X}$. Since equations (5,7) are closed form equations for F, we may obtain a closed form expression for M and hence for $\Sigma_{\hat{\Theta}}$.

4 Testing the Accuracy of the Predicted Covariance Matrix

The matrix $\hat{\Sigma}_{\hat{\Theta}} = M\Sigma_{\hat{X}}M^t$ obtained in the previous section predicts the dispersion of results $\hat{\Theta}$ we would obtain from dispersed values $\hat{X}$ of the pure matches X. In order to check the accuracy of this prediction, we obtained actual values $\hat{\Theta}_1, \ldots, \hat{\Theta}_N$, and compared their dispersion with the predicted covariance $\hat{\Sigma}_{\hat{\Theta}}$. We now describe two methods of comparison.

4.1 Statistical Hypothesis Testing

The first method is to perform an hypothesis test. The hypothesis we are testing is the following:

H_0: *the observations* $\hat{\Theta}_1, \ldots, \hat{\Theta}_N$ *come from a normal random variable* $N(\mu, \Sigma)$, *and* $\Sigma = \beta^2\hat{\Sigma}_{\hat{\Theta}}$ *for some unspecified* β

This hypothesis may be tested by using a likelihood ratio test (see [1] page 262 for details). We first compute the statistic matrix

$$B = \sum_{i=1}^{N}(\hat{\Theta}_i - \bar{\Theta})(\hat{\Theta}_i - \bar{\Theta})^t \tag{10}$$

where $\bar{\Theta} = \sum \hat{\Theta}_i/N$ is the sample mean. Then we compute the ratio

$$\lambda = \frac{(\det(B\hat{\Sigma}_{\hat{\Theta}}^{-1}))^{\frac{N}{2}}}{(\frac{\operatorname{tr}(B\hat{\Sigma}_{\hat{\Theta}}^{-1})}{d})^{\frac{dN}{2}}} \tag{11}$$

where d is the dimension of the samples ($d = 2$ in our case). λ is the maximal likelihood of the observed samples under the assumption H_0, divided by the maximal likelihood of the observed samples under no restrictions. (By the maximal likelihood we mean the likelihood under the choice of μ, Σ that maximizes this likelihood). If this ratio is lower than some threshold λ_0 then we reject the hypothesis H_0.

The threshold λ_0 may be selected to yield significance level α as follows. Under the assumption H_0, the random variable

$$W = \lambda^{N/2} \tag{12}$$

is distributed with cumulative distribution function

$$\Pr(W < w) = w^{\frac{N-2}{2}} \tag{13}$$

Therefore, if we require the hypothesis H_0 to be mistakenly rejected when it is true with probability α, we should choose λ_0 to satisfy

$$(\lambda_0^{2/N})^{\frac{N-2}{2}} = \alpha \tag{14}$$

or

$$\lambda_0 = \alpha^{\frac{N}{N-2}} \tag{15}$$

When the hypothesis is true, an estimate of the scale factor β^2 is given by

$$\beta^2 \approx \frac{\operatorname{tr}(B\hat{\Sigma}_{\hat{\Theta}}^{-1})}{dN} \tag{16}$$

4.2 Geometric Evaluation

The covariance matrix describes through its eigenvectors and eigenvalues the dispersion of a random variable in terms of directions of dispersion and magnitudes of dispersion. In some cases we are able to accurately estimate the direction of dispersion of the FOE estimates, but we cannot expect to get an accurate estimate of the the magnitude of dispersion. This happens when the FOE estimates are scattered on a line. The covariance matrix then is close to singular. For these cases and in addition to the previous test described, we will now describe how to test the quality of our covariance prediction by a direct geometric comparison.

Let $\hat{\Sigma}_{\hat{\Theta}}$ be the predicted covariance matrix, and let Σ_e be the best unbiased estimate of the covariance matrix, obtained from the samples:

$$\Sigma_e = \frac{1}{N-1}\sum_{i=1}^{N}(\hat{\Theta}_i - \bar{\Theta})(\hat{\Theta}_i - \bar{\Theta})^t \tag{17}$$

The equi-(probability density) contours for the FOE are ellipses with axes in the direction of the eigenvectors of the covariance matrix. The lengths of these axes depend on the square root of the eigenvalues (the lower the probability density - the longer the axes). Let $\hat{u}, \vec{u}_e$ be the unit eigenvectors corresponding to the larger eigenvalues of $\hat{\Sigma}_{\hat{\Theta}}, \Sigma_e$ respectively. Let $\hat{\lambda}_1, \hat{\lambda}_2, \lambda_{1e}, \lambda_{2e}$ be the eigenvalues of these matrices. A geometric check of validity of the prediction $\hat{\Sigma}_{\hat{\Theta}}$ is thus to look at the angle between the vectors $\hat{u}$ and $\vec{u}_e$, and at the ratios

$$\sqrt{\frac{\hat{\lambda}_1}{\lambda_{1e}}}, \sqrt{\frac{\hat{\lambda}_2}{\lambda_{2e}}}$$

However, one has to note the following exception. Suppose the dispersion described by the covariance matrix is close to circular. In other words, the value of $\sqrt{\hat{\lambda}_1/\hat{\lambda}_2}$ is not much larger than 1. Then the direction of the principal axis of the ellipse is rather arbitrary. In that case, we should not expect the angle between $\hat{u}$ and $\vec{u}_e$ to be close to zero.

5 Results

Our SUF computation method was tested on synthetic and real images. We created a synthetic 3D scene with 20 points. These points were projected perspectively to yield the base image. The second image was then obtained by projecting these points on a translated screen. The direction and magnitude of the translation vector was varied to show the different behaviour of the sensory uncertainty at different configurations of the robot. The projected points were corrupted with Gaussian noise with standard deviations $\sigma_x = \sigma_y = 2$ pixels (the focal length was taken as 1000).

For each translation of the second screen w.r.t the first screen, we made made 50 different noise corrupted versions of the projected features. For each version we computed the FOE by minimizing the objective function given by equation (7). This gave us the actual dispersed values $\hat{\Theta}_1, \ldots, \hat{\Theta}_{N=50}$. Then we computed the predicted covariance matrix $\hat{\Sigma}_{\hat{\Theta}}$ by using equation (9). (The matrix M was evaluated at an arbitrary sample $\hat{\Theta}_{50}$ and at the measured features $\hat{X}$). We tested the validity of $\hat{\Sigma}_{\hat{\Theta}}$ with respect to the dispersed actual values obtained as described in the previous section.

The coordinate system in which we worked is the standard normalized camera coordinates of the base image - i.e. the z axis is the viewing direction and the x, y axes are on the image plane. In the first case we will present, the second screen was translated along the x (i.e. left/right) axis and the z (i.e. forward/backward) axis, with respect to the base image position. Figure 3 shows those configurations in which the hypothesis H_0 had to be rejected (with significance level $\alpha = 5\%$). It may be seen that apart from the cases where the translation was nearly parallel to the screen, the hypothesis was almost always not rejected. Indeed, apart from middle two lines, the hypothesis is rejected in 23 out of 440 configurations which is very close to 5%. Figure 4 shows the distribution of the estimate given in equation (16) for the scale factor β^2, for those configurations where the hypothesis was not rejected. It may be seen that the scale factor is close to 1, as expected.

In the configurations where the z component of the translations was low, the hypothesis H_0 was rejected. However, the predicted covariance matrix still gives a very good qualitative measure of dispersion. Figure 5 shows the angular difference between the predicted and empirical principal axes of the dispersion ellipses - i.e. the angle between the eigenvectors. It may be seen that in the configurations that seemed in figure 3 to be "problematic", the difference is actually very low. In the configurations with small x translation (i.e. those with FOE near the center of the screen) we now obtain large angular errors between predicted and empirical eigenvectors. This is a result of the circularity of the dispersion in those configurations, as is shown in figure 6. The figure clearly shows that the large angular differences are obtained in cases where the axes of the ellipse are nearly equal , and hence the direction of the principal axis is arbitrary. On the other hand, figure 7 shows the distribution of angular difference between predicted and empirical eigenvectors, for configurations in which the dispersion was not circular: the square root of the ratio of empirical eigenvalues was more than 1.5. It is may be seen that the predicted directions of dispersion are very close to the actual dispersion obtained.

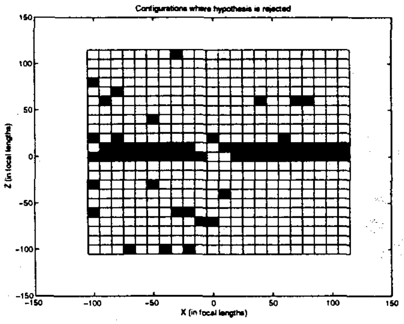

Figure 3: Configurations in which the hypothesis is rejected

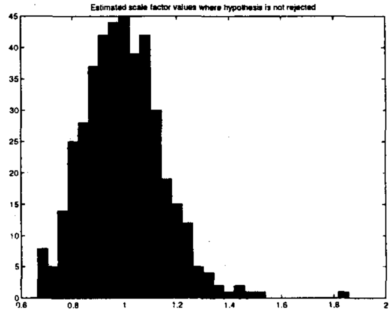

Figure 4: Histogram of scale factor in configurations where the hypothesis is not rejected

In another synthetic scene the second image was obtained by translations with varying x, y components and fixed z component. In other words the camera was moved forward a fixed amount, and then sideways and up and down by varying amounts. We measured the dispersion of the FOE estimate obtained by the product

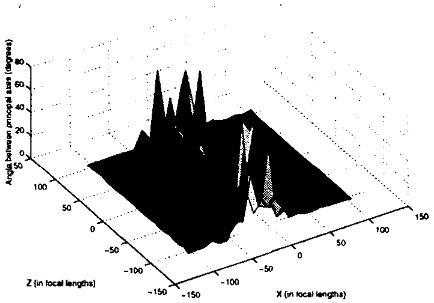

Figure 5: Angle between predicted and empirical eigenvectors (degrees)

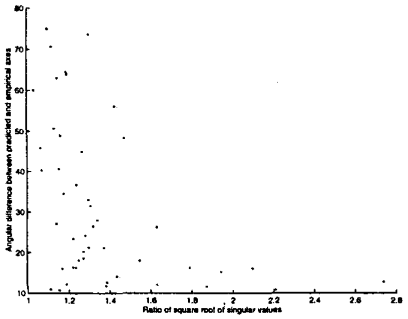

Figure 6: Ratio of axes lengths for configurations where angle between predicted and empirical eigenvectors was larger than 10 degrees

$\sqrt{\lambda_1\lambda_2}$ which is proportional to the area of the ellipse determining the dispersion. Figure 8 shows both the predicted and the actual measures of dispersion that were obtained. It is seen that the prediction is quite accurate.

In addition to various synthetic scenes, we have tested our method on real images. Figure 9 shows the base image and the images obtained after a right and forward translation respectively. Around 30 pairs of corresponding points were found between the base image and each of the other two images. Based on these points, the FOE was computed and its dispersion predicted by our method. These predictions were then compared to the dispersion of FOE values obtained by minimizing the objective function on noised versions of the point correspondences.

Figure 10 shows the actual dispersion of 50 FOE estimates, and the dispersions described by the predicted

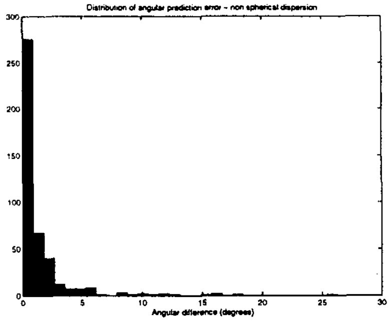

Figure 7: Distribution of angular difference between predicted and empirical eigenvectors, where dispersion is not circular

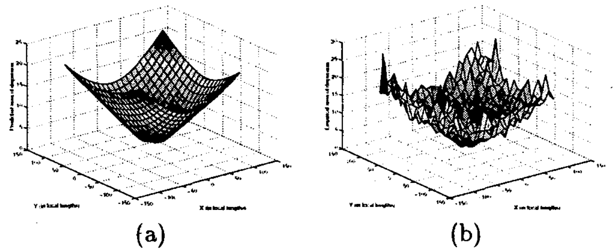

Figure 8: Measure of dispersion in FOE estimate (a) Predicted (b) Empirical

and empirical covariance matrices, for localization after forward translation. The angle between the principal axes of the two ellipses is 11.9 degrees. The ratios of the lengths of axes $(\sqrt{\hat{\lambda}_1/\lambda_{1e}}, \sqrt{\hat{\lambda}_2/\lambda_{2e}})$ are 1.01 and 1.08. As is evident by these numbers and by looking at the figure, the prediction is quite accurate. The likelihood ratio test also confirms the hypothesis with ratio $\lambda = 0.38$ and with estimated scale factor $\beta^2 = 0.92$.

Localization using the second image is the case of localization after sideways translation. In this case the FOE is practically at infinity. The angular difference between the predicted and empirical eigenvectors was 0.08 degrees. The actual empirical dispersion is very high in the direction of the principal axis. The predicted covariance matrix does indeed predict this with singular values ratio of 30000. Thus, in this case, the qualitative behaviour of the dispersion is predicted correctly by our method, and the quantitative behaviour is meaningless.

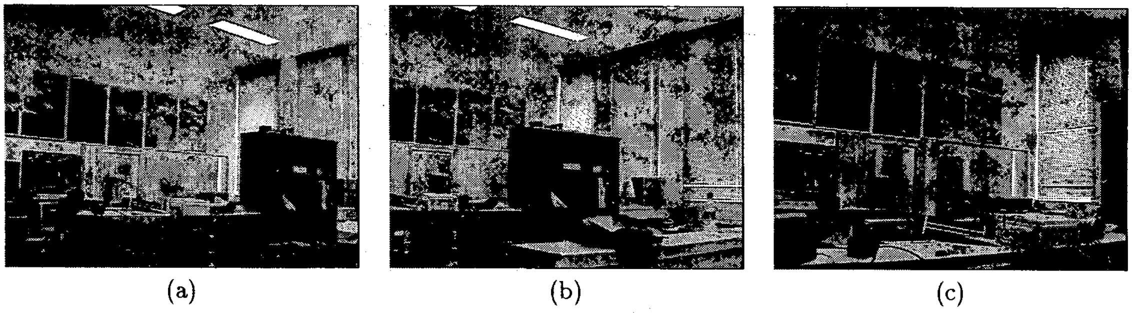

Figure 9: Real images used. (a) Base image (b) Second image - translation parallel to screen (c) Third image - forward translation

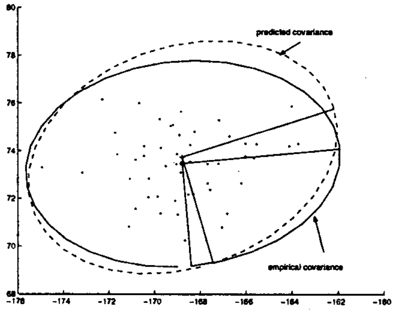

Figure 10: Predicted vs. empirical dispersion

6 Conclusions

In recent years it has been recognized that in order to achieve robust motion planning and navigation algorithms, the varying capability of the sensors the robot is using should be taken into account. Having a mapping of sensor performance across the configuration space has been argued to be beneficial and important. However, despite the importance of vision as a localization sensor, there has been limited work on creating such a mapping for a vision sensor. In this work we have addressed this need.

We have presented a new method to compute the direction of translation of the robot, and have shown that together with the estimated direction one may obtain an indication of the accuracy of this estimate - the covariance matrix. The covariance matrix is computed by a closed form formula and hence its computation is fast. We have shown that the predicted covariance describes accurately the dispersion of estimates in different configurations.

Having a reliable performance map, which describes the quality of the localization result given to us by our sensor, we may now address higher level problems. Future work will integrate the performance map into the motion planning stage: the motion planner should use the data in the map to plan paths along which the sensor is able to give accurate localization results. In addition, sensing strategies can be devised which use the performance map to decide at which points along the path the sensor should be invoked to update the robot's position.

References

[1] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley Publications in Mathematical Statistics, 1958.

[2] R. Basri, E. Rivlin, and I. Shimshoni. Visual homing: Surfing on the epipoles. *International Journal of Computer Vision*, 33(2):117–137, 1999.

[3] T. Celinski and B. McCarragher. Achieving efficient data fusion through integration of sensory perception control and sensor fusion. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1960–1965, 1999.

[4] T. Celinski and B. McCarragher. Improving sensory perception through predictive correction of monitoring errors. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2608–2613, 1999.

[5] T. Fraichard and R. Mermond. Path planning with uncertainty for car-like robots. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 27–32, 1998.

[6] R. M. Haralick. Propagating covariance in computer vision. In *Proc. of 12'th ICPR*, pages 493–498, 1994.

[7] A. Lambert and N. L. Fort-Piat. Safe actions and observation planning for mobile robots. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1341–1346, 1999.

[8] N. Roy, W. Bugard, D. Fox, and S. Thrun. Coastal navigation - mobile robot navigation with uncertainty in dynamic environments. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 35–40, 1999.

[9] R. Sharma and H. Sutanto. A framework for robot motion planning with sensor constraints. *IEEE Transactions on Robotics and Automation*, 13(1):61–73, February 1997.

[10] H. Takeda, C. Facchinetti, and J.C. Latombe. Planning the motions of a mobile robot in a sensory uncertainty field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1002–1017, October 1994.

[11] S. Thrun. Finding landmarks for mobile robot navigation. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 958–963, 1998.

[12] M. Werman and D. Keren. A Bayesian method for fitting parametric and non-parametric models to noisy data. In *Proc. of IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, June 1999.