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

Amit Adam, Ehud Rivlin, and Ilan Shimshoni, *Member, IEEE*

***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 present a method for computing the performance map of a vision-based sensor. We compute the map and show that it accurately describes the actual performance of the sensor, both on synthetic and real images. The method we use (following [10]) 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.**

***Index Terms*—Motion planning, performance evaluation, uncertainty, vision-based localization.**

I. Introduction

EXTERNAL sensors such as video cameras, laser range finders, and sonar are being routinely used for mobile robot localization. Recently, [9], [11], [13], [14], [15], [16] 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 [15].

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 [9], [11], [12], [15]. In [13] a related notion is the information content of the environment at each configuration. Another similar idea motivated by visual servoing is described in [14]. Another use of the SUF map is demonstrated in [16], where the question of choosing the proper landmarks for localization is addressed. Celinski and McCarragher [5], [6] 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 [11], [13], [15] 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 [13], [15].

In this paper, we present a fast method to compute a SUF for a vision sensor. Our 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 our 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 on both synthetic and real images.

In the next section we describe the localization algorithm based on input from a vision sensor. Afterward, we describe the general method to estimate the covariance matrix of the sensor output, and apply it to our specific case. Section IV describes how we statistically tested the validity of our predicted covariance matrix. In Section V we present comparison of the predicted vs. actual covariance matrices obtained in different scenes and configurations. The last section summarizes and concludes this paper.

II. Localization Algorithm

Vision has been used as a sensor for navigation in many different ways (a small sample of examples is [1], [7], [8]). In this work we use vision by computing the motion of a camera (known as egomotion recovery) which is mounted on the robot. A base image is taken at a known configuration of the robot and camera. When localization is required, a second image is obtained. Point correspondences between the two images are found. From these correspondences the rotation and the direction of translation of the camera (w.r.t the base image configuration) may be found. In order to estimate the magnitude of translation, the robot has to make a small move and obtain a third

Manuscript received July 4,2000. This paper was recommended for publication by Associate Editor E. Pagello and Editor S. Hutchinson upon evaluation of the reviewers' comments.

A. Adam is with the Department of Mathematics, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: amita@tx.technion.ac.il).

E. Rivlin is with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: ehudr@cs.technion.ac.il).

I. Shimshoni is with the Department of Industrial Engineering and Management, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: ilans@ie.technion.ac.il).

Publisher Item Identifier S 1042-296X(01)06732-5.

image. Since the camera is mounted on the robot, and since the base image configuration of the robot and camera are known, the current configuration of the robot may now be deduced.

Our goal is now to predict the covariance matrix of the motion estimate we obtain. We will focus on the estimate of direction of translation, or focus of expansion (FOE). Note that the method which we use (following [10]) is rather general and may also be used in the case where a rotation exists.

A. Maximum Likelihood Formulation

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 that 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

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

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

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

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

This definition is illustrated in Fig. 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 [17] for related work in which actual computation of simpler, one dimensional integrals resembling the integral in (2), was carried out in the context of curve fitting).

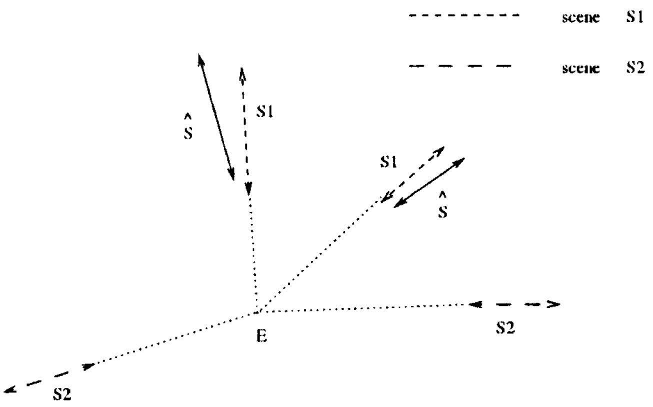


Fig. 1. Possible supporting scenes for the point E.

B. Approximating the Maximum Likelihood Estimator

Since the exact computation of the likelihood function introduced in (2) may be difficult, we now introduce an approximation. The exact value of the likelihood function (2) is obtained by integrating the likelihood of various scenes. We chose to approximate this value by taking only the largest integrand into account. Let us define for a given point (u, v) the scene $\tilde{S}$ to be the scene that supports (u, v) and that is also the most likely scene given $\hat{S}$ (with respect to all other scene supporting (u, v)):

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

We now define the function q to be

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

In other words, the value $q(u, v)$ is the largest value being integrated in (2).

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.

In order to compute the function $q(u, v)$, 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 $\hat{S}$. 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 Fig. 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 γ with the x axis such that

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

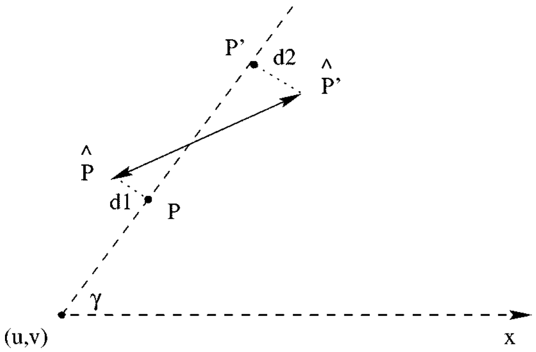


Fig. 2. Definition of the function q.

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 $\gamma = \gamma(u, v, \hat{x}, \hat{y}, \hat{x}', \hat{y}')$ the distances d_1, d_2 to the line are

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

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

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

C. Further Remarks

The approach for computing the FOE we have chosen is different from the standard least-squares approach. In the standard approach one searches for the point which is closest to the lines passing through the matching points. The lines passing through the pairs of matching points are treated as “ground truth”. Since they are not the ground truth they do not all meet at one point. The FOE is taken to be the best compromise on this requirement—i.e., the point which has the minimal sum of squared distances from these lines.

On the other hand, in our approach the lines passing through the pairs of matching points are not taken as they are. They are allowed to be changed since we recognize that the points defining these lines are noisy. Fig. 3 shows an example which demonstrates the effect of this difference in approaches. In the figure, we see five pairs of points which are noise less. At the bottom right there is a pair of points in which one point was corrupted. The correct FOE is at the origin. The sixth line coming from the noisy pair distorts the least-squares estimate of the FOE since this line is taken as is. In our approach, the sixth line is not taken as is and therefore our estimate of the FOE is much less distorted.

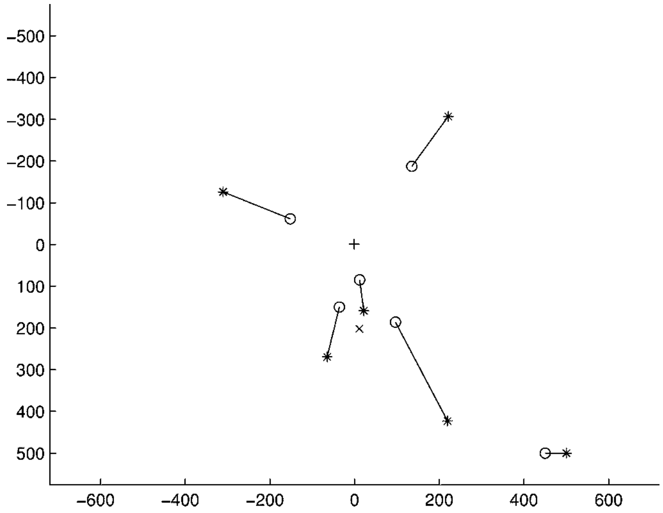


Fig. 3. An example showing the difference between the suggested approach and the least squares approach for estimation of the FOE. The + marks the AML FOE and the x marks the least-squares FOE. The true FOE is in the origin.

We stress the fact that the AML estimate for the FOE is *not* a robust estimate. Indeed, even a single outlier matching pair may distort our FOE estimate. The analysis of robustness of this estimator is omitted for reasons of space.

As a final remark, let us consider the case in which the translation is parallel to the image plane. In this case the segments in the scene are parallel and the FOE is at infinity. It may be shown (again the details are omitted for reasons of space) that if the point (u, v) moves toward infinity on a straight line, the function F will tend to a limit L. The limit L will be equal to zero if and only if the straight line is parallel to the line segments. If the line is not parallel to the segments the limit L will be strictly positive. Hence, minimization of the function F will lead us to infinity as desired.

III. 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 [10]. 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 [10] 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 $\hat{\Theta}$:

$$\mathcal{C}_{\hat{\Theta}} = M\mathcal{C}_{\hat{X}}M^t \tag{9}$$

where

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

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 (5) and (7) are closed-form equations for F, we may obtain a closed-form expression for M and hence for $\mathcal{C}_{\hat{\Theta}}$.

IV. Testing the Accuracy of the Predicted Covariance Matrix

The matrix $\hat{\mathcal{C}}_{\hat{\Theta}} = M\mathcal{C}_{\hat{X}}M^t$ 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. We would like to check the quality of this prediction of dispersion. In order to do this, we obtained actual values $\hat{\Theta}_1,\ldots,\hat{\Theta}_N$, and compared their actual dispersion with the predicted covariance $\hat{\mathcal{C}}_{\hat{\Theta}}$. The comparison is done by two different methods which will now be described.

A. Statistical Hypothesis Testing

The first method is to perform an hypothesis test. We assume that the actual dispersed values $\hat{\Theta}_1,\ldots,\hat{\Theta}_N$ come from a normal distribution $N(\mu,\mathcal{C})$, and hypothesize about the covariance matrix $\mathcal{C}$. We then employ a statistical test to check whether the hypothesis should be accepted or rejected. The hypothesis we test is as follows:

H_0: *The covariance matrix* $\mathcal{C}$ *satisfies* $\mathcal{C} = \beta^2\hat{\mathcal{C}}_{\hat{\Theta}}$ *for some unspecified* β

Note that we do not hypothesize anything about the mean of the distribution.

This hypothesis may be tested by using a likelihood ratio test (see [4] 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

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

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 $\mu,\mathcal{C}$ 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

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

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

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

or

$$\lambda_0 = \alpha^{N/(N-2)} \tag{15}$$

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

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

B. Geometric Evaluation

We now present a second approach to testing the quality of our predictions. Before describing the approach, we will list a few reasons why we should use this approach in addition to the statistical tests described previously.

The first reason is that this approach is more direct and intuitive. As will be shown, we directly compare the predicted and actual geometries. A second reason is that we have not formally justified why the FOE values are normally distributed (although this will be shown by the experimental results). This assumption is required for the previous test to be employed.

The third reason follows such that 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 reasons 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 $\hat{\mathcal{C}}_{\hat{\Theta}}$ be the predicted covariance matrix, and let $\mathcal{C}_e$ be the best unbiased estimate of the covariance matrix, obtained from the samples:

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

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 $\hat{\mathcal{C}}_{\hat{\Theta}}$, $\mathcal{C}_e$, respectively. Let $\hat{\lambda}_1$, $\hat{\lambda}_2$, λ_{1e}, λ_{2e} be the eigenvalues of these matrices. A geometric check of validity of the prediction $\hat{\mathcal{C}}_{\hat{\Theta}}$ 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.

V. Results

Our SUF computation method was tested on synthetic and real images. We created a synthetic 3-D 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 behavior 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 50 different noise corrupted versions of the projected features. For each version we computed the FOE by minimizing the objective function given by (7). This gave us the actual dispersed values $\hat{\Theta}_1, \ldots, \hat{\Theta}_{N=50}$. Then we computed the predicted covariance matrix $\hat{\mathcal{C}}_{\hat{\Theta}}$ by using (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 $\hat{\mathcal{C}}_{\hat{\Theta}}$ 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. Fig. 4 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%. Fig. 5 shows the distribution of the estimate given in (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.

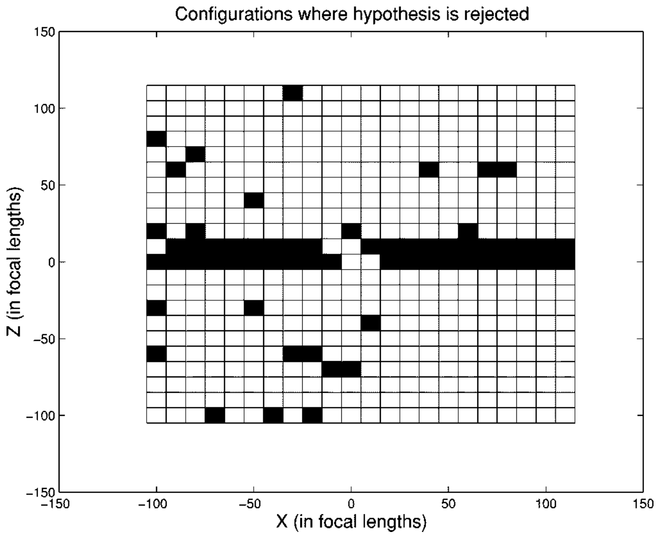


Fig. 4. Configurations in which the hypothesis is rejected.

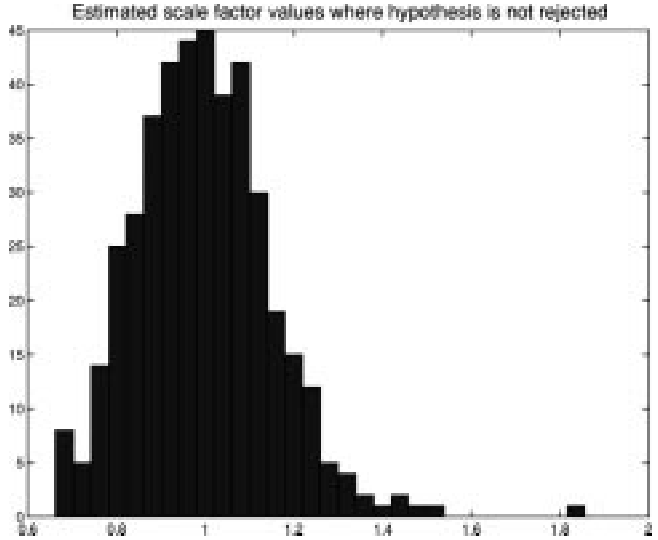


Fig. 5. Histogram of scale factor in configurations where the hypothesis is not rejected.

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. Fig. 6 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 Fig. 4 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

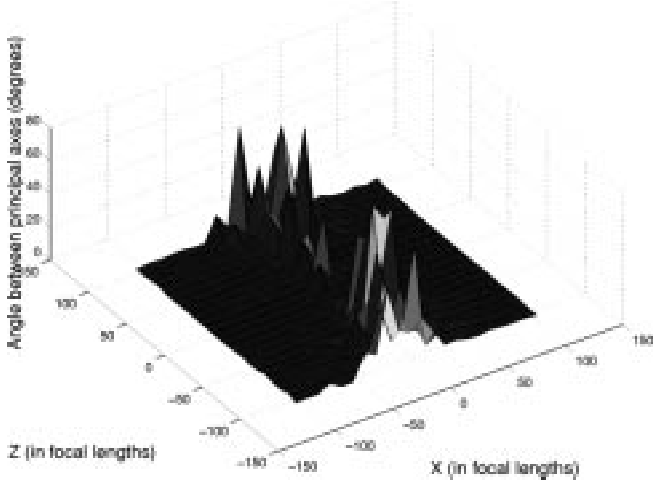


Fig. 6. Angle between predicted and empirical eigenvectors (degrees).

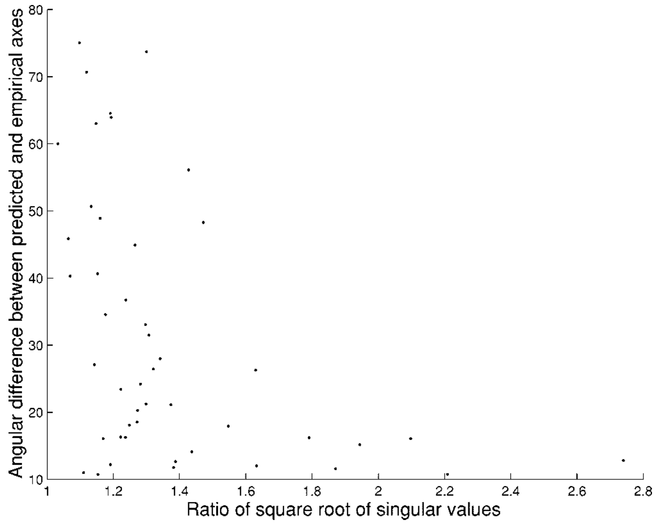


Fig. 7. Scatter plot of angular difference and circularity of dispersion. Large angular differences occur when the dispersion is circular—i.e., the ratio of axes lengths is close to 1. We have plotted only the cases in which the angular difference was more than 10°.

the circularity of the dispersion in those configurations, as is shown in Fig. 7. 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, Fig. 8 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.

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 $\sqrt{\lambda_1\lambda_2}$ which is proportional to the area of the ellipse determining the dispersion. Figs. 9 and 10 show the predicted and actual measures of dispersion obtained. It is seen that the prediction is quite accurate. (We will discuss later the significance of the shape of the graph shown in these figures.)

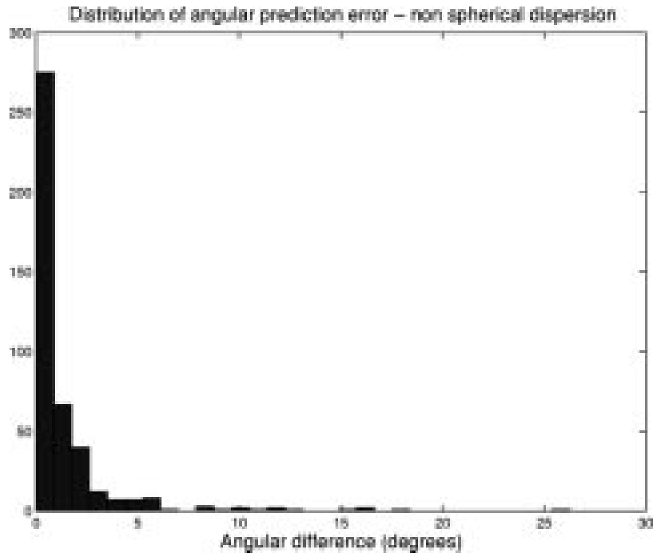


Fig. 8. Distribution of angular difference between predicted and empirical eigenvectors, where dispersion is not circular.

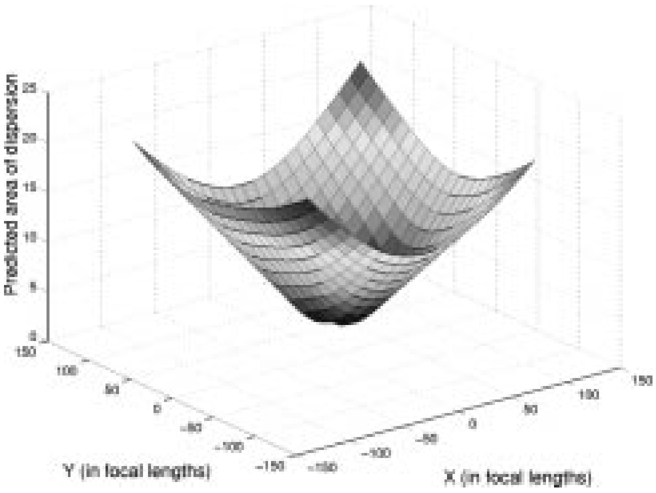


Fig. 9. Predicted measure of dispersion.

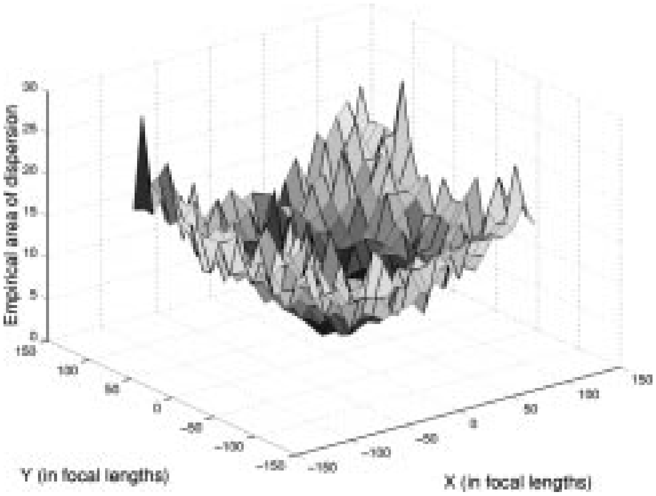
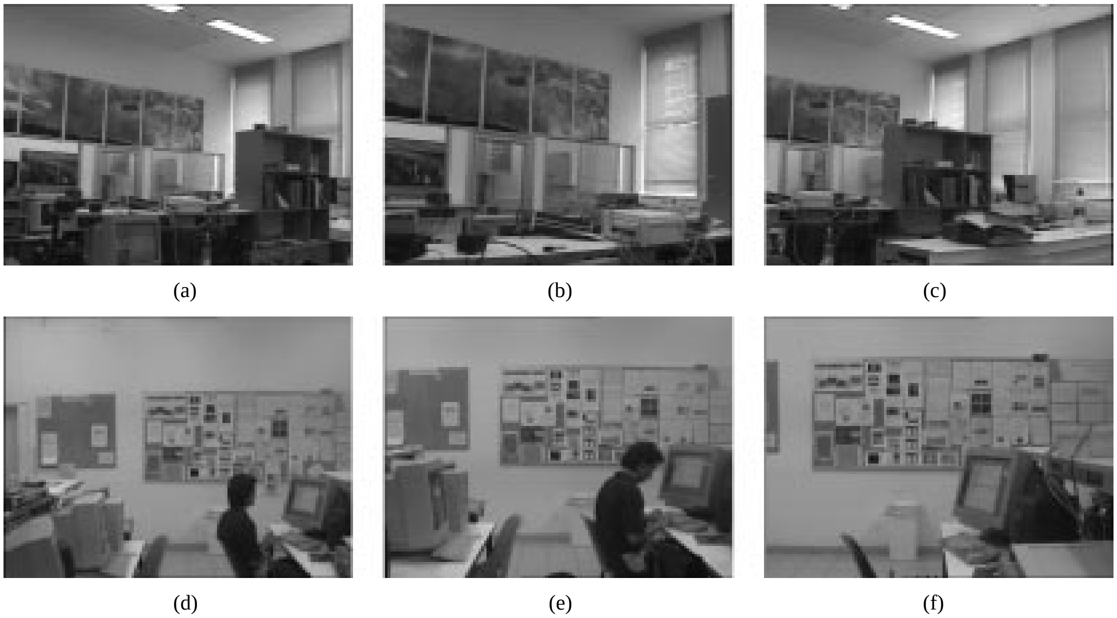


Fig. 10. Empirical measure of dispersion obtained.



(a) (b) (c)

(d) (e) (f)

Fig. 11. Real images used. (a) Base image of first set. (b) Image after forward translation. (c) Image after translation parallel to screen. (d) Base image of second set. (e) Image after forward and downward translation. (f) Image after forward, sideway, and downward translation.

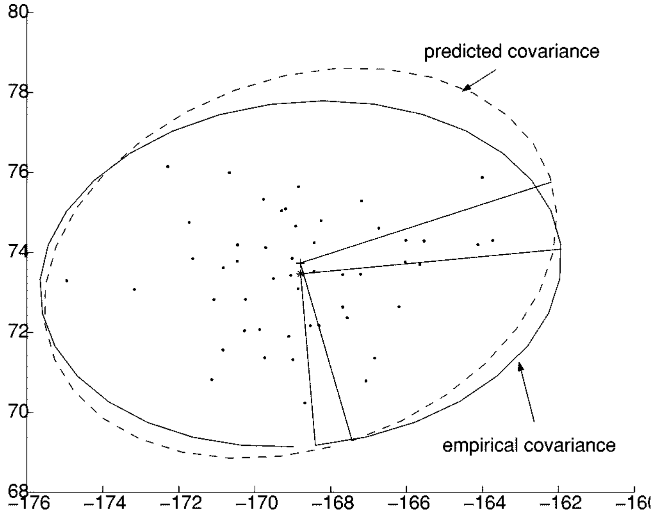


Fig. 12. Predicted versus empirical dispersion—images (a) and (b).

In addition to various synthetic scenes, we have tested our method on real images. Two sets of images were taken in our lab. In each set we have a base image, and two other images which were obtained after different translations with respect to the base image. The two sets of images are shown in Fig. 11.

For each pair of images consisting of a base image and one of the translated images, around 30 pairs of corresponding points were found. 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.

Fig. 12 shows the actual dispersion of 50 FOE estimates, and the dispersions described by the predicted and empirical covariance matrices, for the first pair of images [images (a) and (b)]. The angle between the principal axes of the two ellipses is 11.9°. The ratios of the lengths of axes

$$\left(\sqrt{\frac{\hat{\lambda}_1}{\lambda_{1e}}}, \sqrt{\frac{\hat{\lambda}_2}{\lambda_{2e}}}\right)$$

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 image (c) and image (a) 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°. 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 30 000. Thus, in this case, the qualitative behavior of the dispersion is predicted correctly by our method, and the quantitative behavior is meaningless.

Figs. 13 and 14 present comparisons of the empirical and predicted covariance matrices for the image pairs (d), (e), and (d)–(f), respectively. We drew the ellipses defined by the empirical and predicted covariance matrices, and the actual FOE estimates that were obtained (based on which the empirical covariance matrix was computed). For the pair (d) and (e), the angle between the principal axes of the ellipses is 1.2°. The ratios of lengths of the axes are 0.77 and 1.1. We can see that the direction of dispersion is predicted very well although the length of the principal axis was under-predicted. For the pair (d)–(f), the angle between the principal axes is 1 degree. The ratio of lengths of the axes are 1.17 and 1.23.

A. Angular Errors

Recall that our goal is localization of a robot based on egomotion estimation. Assume that the FOE is (u, v). Then in the 3–D

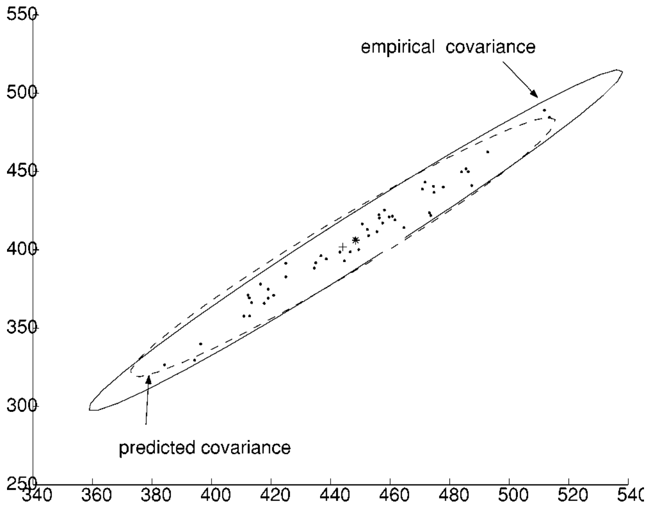


Fig. 13. Predicted versus empirical covariance for images (d) and (e).

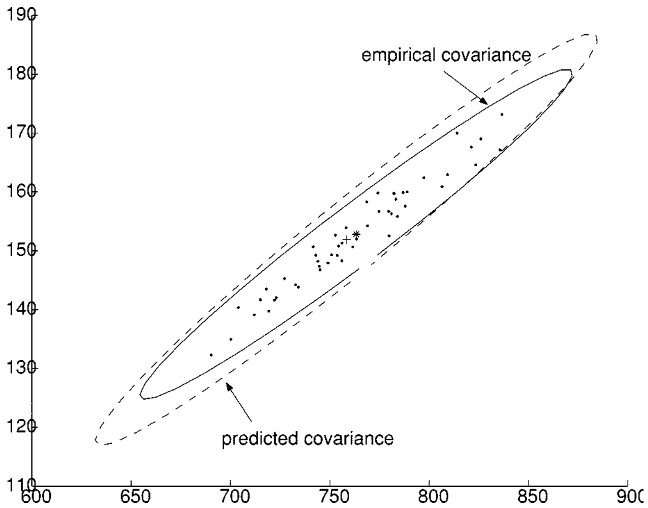


Fig. 14. Predicted versus empirical covariance for images (d)–(f).

world, the direction of translation was $\vec{d} = (u, v, f)^t$ (where f is the focal length). We will describe this 3-D direction using the pan and tilt angles. By tilt we mean the angle that is created by the v component and by pan we mean the angle that is created by the u component. Explicitly written, the direction of the vector $(u, v, f)^t$ is specified by

$$\begin{pmatrix} \theta \\ \varphi \end{pmatrix} = g(u, v) = \begin{pmatrix} \arctan \frac{u}{f} \\ \arctan \frac{v}{f} \end{pmatrix}.$$

Let $\mathcal{C}_{u,v}$ denote the covariance matrix of the FOE estimate $(\hat{u}, \hat{v})$. Let $(\hat{\theta}, \hat{\varphi})^t = g(\hat{u}, \hat{v})$ be the corresponding angles representing the estimated 3–D direction. Let $\mathcal{C}_{\theta,\varphi}$ be the covariance matrix of the estimated angles $(\hat{\theta}, \hat{\varphi})^t$. Since we may approximate

$$\begin{pmatrix} \hat{\theta} \\ \hat{\varphi} \end{pmatrix} = g(\hat{u}, \hat{v}) \approx g(u, v) + \frac{\partial g}{\partial (u, v)} \begin{pmatrix} \hat{u} - u \\ \hat{v} - v \end{pmatrix}$$

we may use the following formula for $\mathcal{C}_{\theta,\varphi}$:

$$\mathcal{C}_{\theta,\varphi} = J \mathcal{C}_{u,v} J^t$$

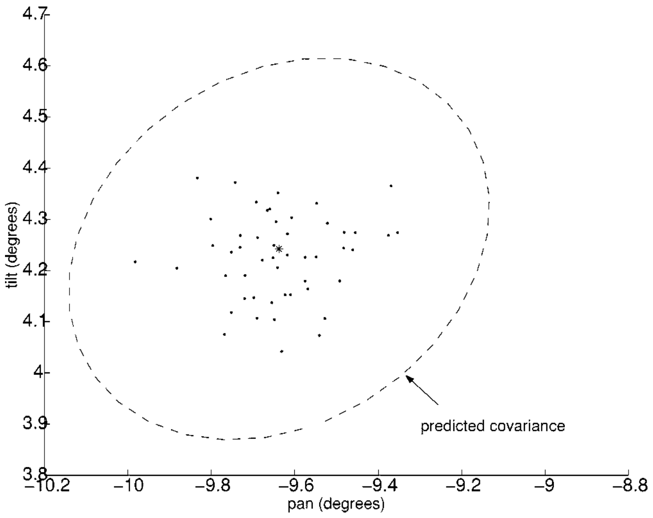


Fig. 15. Pan/tilt dispersion and predicted covariance for images (a) and (b).

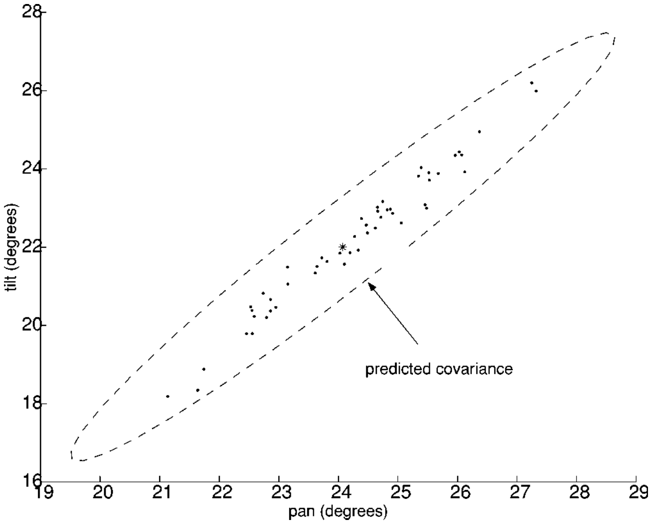


Fig. 16. Pan/tilt dispersion and predicted covariance for images (d) and (e).

where

$$J = \frac{\partial g}{\partial (u, v)} = \begin{pmatrix} \frac{f}{u^2+f^2} & 0 \\ 0 & \frac{f}{v^2+f^2} \end{pmatrix}$$

The matrix $\mathcal{C}_{\theta,\varphi}$ describes the dispersion of the pan and tilt angles which denote the 3–D direction of translation.

We have computed the pan and tilt estimates of the direction of translation for the image pairs (a), (b) and (d), (e). We have also computed the predicted covariance matrices $\mathcal{C}_{\theta,\varphi}$. In Figs. 15 and 16, we plot the actual pan/tilt estimates and the dispersion described by the predicted covariance matrices $\mathcal{C}_{\theta,\varphi}$. It may be seen that the covariance matrices describe the dispersion faithfully.

Describing the dispersion in angular terms may change our view of the uncertainty associated with different configurations. Let us analyze the 3–D error in the synthetic scene for which Fig. 9 was computed. Recall that in Fig. 9 we drew the area of the ellipse containing the FOE with some fixed probability, as a function of the translation parameters. This area is in pixels

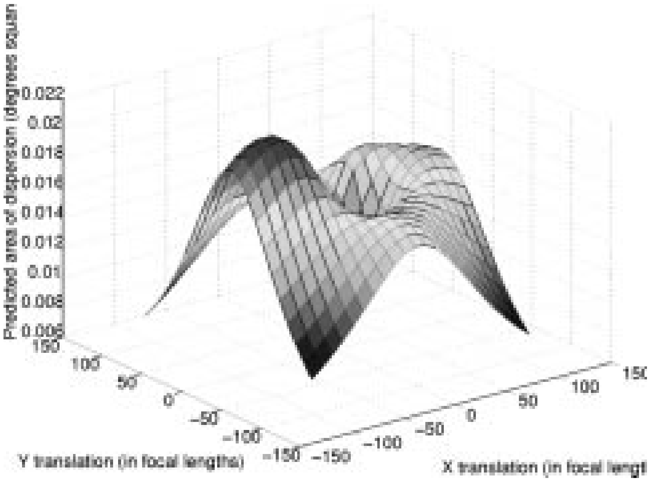


Fig. 17. Dispersion of pan and tilt angle as function of translation. We plotted $\sqrt{\lambda_1\lambda_2}$ for the matrix $\mathcal{C}_{\theta,\varphi}$ corresponding to each translation. Compare with Fig. 9.

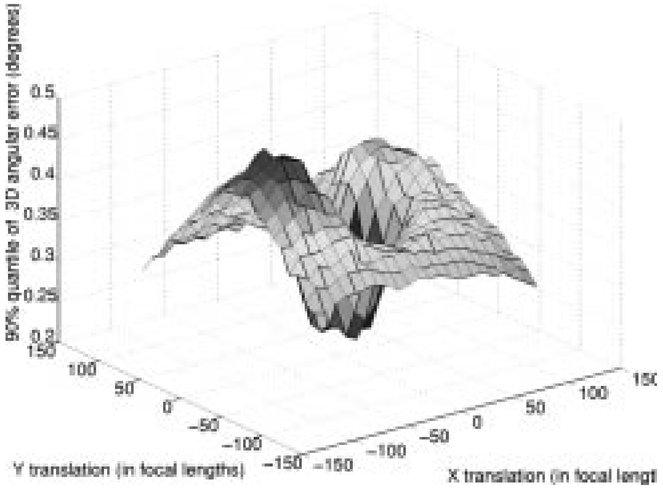


Fig. 18. Empirical measure of dispersion. The graph shows the 90'th percent quantile of the 3–D angular error distribution at each configuration. Compare with theoretical prediction in Fig. 17.

squared. It was seen that as the X, Y components of the translation grow, the area of the ellipse grows. In other words the uncertainty in the location of the FOE is larger the further it is from the origin. Let us now consider the uncertainty in pan and tilt angles. Fig. 17 describes the area of the ellipse containing the true pan and tilt angles which describe the direction of translation, with some fixed probability. This area is in degrees squared. We now see two properties:

1) The dispersion is bounded even for large translations.
2) The dispersion is low for small translations.

The reason for the first property is quite clear. As the X, Y components of the translation grow, the FOE moves further away from the origin. Then even large errors in its determination do not affect the 3-D direction. For example, the points $(500, 30)$ and $(600, 40)$ represent two rather different points when considered as points in the (u, v) plane. However, the 3-D directions $(500, 30, 1)$ and $(600, 40, 1)$ are almost identical.

The reason for the low dispersion when the translations are small is that the dispersion in terms of (u, v) coordinates is very low (see Fig. 9). Therefore the dispersion in terms of angles is also low.

Finally, we will compare the angular dispersion predicted in Fig. 17 with the following empirical measure of dispersion. For each translation we have randomly chosen 5000 points $(\hat{u}_i, \hat{v}_i)$ around the estimate $(\hat{u}, \hat{v})$, by using the predicted covariance matrix. Thus we have obtained 5000 representative results of estimated FOEs for each translation. We then computed the angle between the vectors $(\hat{u}_i, \hat{v}_i, f)^t$ and the vector $(\hat{u}, \hat{v}, f)^t$. These 5000 angles are a sample of the true 3–D angular error distribution associated with each specific configuration. For each configuration we now computed the 90% quantile of the distribution. This is an empirical measure of the dispersion of 3–D angular error as a function of the configuration. Fig. 18 shows these empirical results. These results are in very good agreement with the angular dispersion prediction shown in Fig. 17.

As a last remark, we stress the fact that although our method of estimating the FOE involves minimization of a function of two variables, the time it takes is usually very short. Typically a small number of iterations is sufficient to bring us very close to the solution. In (slow, interpreted) MATLAB the minimization usually requires less than 2 s, and may be speeded easily by using a better starting point.

VI. Discussion and Conclusion

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 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.

The work presented here is a step toward performance prediction for vision-based sensors. However, some obstacles still need to be overcome when trying to predict the *overall* performance of vision sensors. One example is the issue of false matches. It is well known that automatic correspondence of points in two images is a very hard task. Outliers in the corresponding pairs of points pose a problem to any algorithm which is based on matching pairs. In another work [2] we have addressed this problem by developing an algorithm that rejects a significant percent of the outliers (while keeping the inliers). Nevertheless, exact quantification of the ability to produce a set of correct matches still remains an open issue.

Another issue for further research is the prediction of the SUF when the features in the environment are not known beforehand. In this work we based the prediction of uncertainty on the scene of line segments from which the FOE is computed. This approach is beneficial for online estimation of the uncertainty. It would be interesting to extend this work to deal with scenes of segments that are not completely given in advance. For example, we may be given the distribution of depths of features and from this we may infer about the distribution of scenes.

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. The problem that interests us is how to choose a combination of motion planner and sensor for a given environment in order to achieve a robust navigation capability. The sensor is abstractly represented by the SUF it defines. In [3] we show how we can compare different motion planners in term of their suitability for a given combination of environment and SUF defined on that environment. This ability to measure the performance of a given combination of sensor and motion planner in a given environment, may lead to customization of sensors and/or motion planners based on the specific characteristics of the environment.

REFERENCES

[1] A. Adam, E. Rivlin, and H. Rotstein, “Fusion of fixation and odometry for vehicle navigation,” *IEEE Trans. Syst. Man C ybern. A*, vol. 29, pp. 593–603, 1999.

[2] ——, “ROR: Rejection of outliers by rotations in stereo matching,” in *Proc. IEEE Conf. Comput. Vision Pattern Recog. (CVPR)*, June 2000, pp. 2–9.

[3] ——, “Toward a meta motion planner,” in Proc. IEEE Int. Conf. Robot. Automat.(ICRA), 2001.

[4] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, New York: Wiley, Mathematical Statistics, 1958.

[5] T. Celinski and B. McCarragher, “Achieving efficient data fusion through integration of sensory perception control and sensor fusion,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 1999, pp. 1960–1965.

[6] ——, “Improving sensory perception through predictive correction of monitoring errors,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 1999, pp. 2608–2613.

[7] F. Chenavier and J. L. Crowley, “Position estimation for a mobile robot using vision and odometry,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 1992, pp. 2588–2593.

[8] G. Dudek and D. Jugessur, “Robust place recognition using local appearance based methods,” in *Proc. IEEE Int. Conf. Robot. Automat.*, Apr. 2000.

[9] T. Fraichard and R. Mermond, “Path planning with uncertainty for car-like robots,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 1998, pp. 27–32.

[10] R. M. Haralick, “Propagating covariance in computer vision,” in *Proc. 12th ICPR*, 1994, pp. 493–498.

[11] A. Lambert and N. L. Fort-Piat, “Safe actions and observation planning for mobile robots,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 1999, pp. 1341–1346.

[12] A. Lazanas and J.-C. Latombe, “Motion planning with uncertainty: a landmark approach,” *Artif. Intell.*, vol. 76, pp. 287–317, 1995.

[13] N. Roy, W. Bugard, D. Fox, and S. Thrun, “Coastal navigation—mobile robot navigation with uncertainty in dynamic environments,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 1999, pp. 35–40.

[14] R. Sharma and H. Sutanto, “A framework for robot motion planning with sensor constraints,” *IEEE Trans. Robot. Automat.*, vol. 13, pp. 61–73, Feb. 1997.

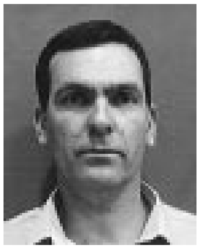
[15] H. Takeda, C. Facchinetti, and J. C. Latombe, “Planning the motions of a mobile robot in a sensory uncertainty field,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 10, pp. 1002–1017, October 1994.

[16] S. Thrun, “Finding landmarks for mobile robot navigation,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 1998, pp. 958–963.

[17] M. Werman and D. Keren, “A Bayesian method for fitting parametric and nonparametric models to noisy data,” in *Proc. IEEE Comput. Vision Pattern Recog. Conf. (CVPR)*, June 1999.

Amit Adam received the B.Sc. degree (*summa cum laude*) from Ben-Gurion University, Beer-Sheva, Israel, the M.Sc. degree from the Weizmann Institute of Science, Rehovot, Israel, both in mathematics, and the Ph.D. degree from the Technion, Israel Institute of Technology, for his work on vision-based robotic navigation, which was performed in the Intelligent Systems Laboratory of the Computer Science Department.

His main research interests are computer vision and robotic navigation.



Ehud Rivlin received the B.Sc. and M.Sc. degrees in computer science, the M.B.A degree from the Hebrew University, Jerusalem, and the Ph.D. degree from the University of Maryland, College Park.

Currently, he is an Associate Professor with the Computer Science Department, Technion—Israel Institute of Technology. His research interests are in machine vision and robot navigation.

Ilan Shimshoni (M’91) recieved the B.Sc. degree in mathematics from the Hebrew University, Jerusalem, in 1984, the M.Sc. degree in computer science from the Weizmann Institute of Science, Rehovot, Israel, in 1989, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign in 1995.

He was a Postdoctoral Fellow with the faculty of Computer Science, Technion—Israel Institute of Technology, from 1995 to 1998, and joined the faculty of Industrial Engineering and Management, Technion, in 1998. His main research interests are in the fields of computer vision and robotics and especially topics that lie on the border between the two fields, such as visual navigation of robots.