

# Using Model-Based Localization with Active Navigation

Amit Adam  
Dept. of Mathematics  
amita@tx.technion.ac.il

Ehud Rivlin  
Dept. of Computer Science  
ehudr@cs.technion.ac.il

Ilan Shimshoni  
Dept. of Industrial Engineering  
ilans@ie.technion.ac.il

Technion – Israel Institute of Technology  
Haifa 32000 – Israel

## Abstract

*Vision is an important sensor used for mobile robot navigation. One approach to localization which is based on vision is to compute camera egomotion with respect to base images. What characterizes this method of localization is that its performance varies greatly in different positions. Active navigation is an approach to path and sensing planning which is designed to address varying performance of a sensor across the configuration space. In this paper we describe how to integrate a vision-based localization sensor with active navigation. We explain the localization process, how its performance varies across the configuration space, and the use of this variation by active navigation.*

## 1 Introduction

The ability to navigate from one point to another is a fundamental requirement for an autonomous mobile agent. For mobile robots, odometry is a basic navigation tool. However because odometry errors accumulate over time, it is common practice to augment the robot with external sensors which are used for navigation. Among these, vision is an important sensor. In this paper we present the use of a vision sensor within the *active navigation* paradigm.

One of the ways to use vision as a navigation sensor is to compute camera egomotion between images taken from the current configuration, and a base image taken from a fixed, known configuration. One advantage of this method is that it uses a small number of base images. However, for a fixed base image, at different positions of the robot we expect this positioning method to have different levels of success. In some positions with respect to base image the method might fail (because of occlusions for example), and in other positions it will work but with different levels of accuracy.

This characteristic of the localization sensor motivates us to use it in conjunction with active navigation. Active navigation [1] is an approach to navigation which takes into account the varying performance level

of a positioning sensor across the configuration space. In this approach we consider the performance of the sensor together with two other important factors: the developing uncertainty in position, and the required level of localization. We use all these factors to plan sensing operations and actions which will enable the robot to navigate along its path.

In the next section we will motivate and describe the active navigation paradigm. We then describe our vision based localization method and how we estimate its different levels of performance at different configurations. Then we describe how we may optimally plan sensing operations along the path. The last section concludes the paper.

## 2 Active Navigation

### 2.1 Background and Motivation

In previous works [12, 6, 8, 10, 13, 11] it has been recognized that the accuracy of the localization obtained by invoking a 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. It is then natural to try and plan the sensing operations to occur while the robot is in areas in which the sensors work well: according to the above mentioned map, the localization quality in these areas is good.

Works which have considered this idea may be found in [12, 6, 8]. In these works the motion planning algorithm uses the sensor performance map to plan paths. In [10] a related notion is the information content of the environment at each configuration. Another similar idea motivated by visual servoing is described in [11].

The motion planning approach taken in previous works (for example [12, 10]) is to search for a path in free space by minimizing a function which takes into account both the length of the path and the sensory uncertainty along the path. (The sensor performance

map has been termed Sensory Uncertainty Field (SUF) in [12]). This approach involves some arbitrary decision on how to trade off between sensory uncertainty and path length. These two different factors are usually combined into one objective function by introducing an arbitrary scale factor between the two.

We suggest an alternative approach. In our approach we first consider the “nominal path” the robot has planned by using a specific motion planning algorithm. Had there been no practical problems such as odometry errors and inaccurate prior information, this “nominal path” is the path the robot would have executed. We then consider the localization accuracy required along the path. This accuracy is determined by the special characteristics of the path and the motion planner which generated the path. This required localization accuracy, in addition to the sensory uncertainty field, affect our decisions on where the sensor should be invoked in order to update the position of the robot.

To summarize, we may say that previous approaches strive for the highest localization accuracy possible and compromise on this accuracy in order to account for the length of the path. In our approach we strive for localization accuracy which is at the level required by the nominal path which was generated by the motion planner. Actions are needed by the robot in order to ensure that at all times the localization accuracy which is achieved by the robot is in agreement with the required localization accuracy.

## 2.2 The Factors Involved

Let us assume a motion planning algorithm (see for example [9]) has planned a path between the source configuration and the target configuration. Our robot intends to execute this “nominal” path. With the environment and the path we associate three factors: the uncertainty in configuration, the accuracy required in the answer obtained after a position (or configuration) update, and the level of accuracy in localization which is obtainable by using the sensor in the current configuration. We now elaborate on each of these factors.

## 2.3 Uncertainty in Configuration

As the robot moves it keeps track of its current configuration. For various reasons (see for example [5]) uncertainty in the current configuration develops. We model this uncertainty as a probability distribution on the configuration space. Let  $U(t)$  be the probability distribution representing the configuration of the robot at time  $t$  along the nominal path.

Different models have been suggested for the development of uncertainty in configuration. For example we may use the Gaussian probability distribution to

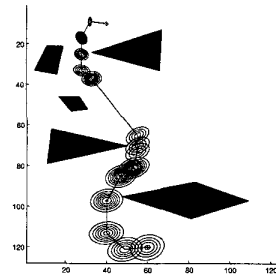


Figure 1:  $U(t)$  - probability distribution of configuration at time  $t$  along the nominal path

model the position of the robot. The mean of the distribution is the nominal final position. The covariance matrix describes dispersion which is proportional to the length of the motion command. Its principal eigenvector is in the direction perpendicular to the heading direction, and the other eigenvector is in the heading direction. Fig. 1 shows an example of the development of uncertainty along the nominal path. A different model based on the triangular distribution may be found in [14].

## 2.4 Accuracy Requirement

We now consider the level of accuracy in position that is required from the sensor being used for position update. We consider the localization accuracy required *in the context of performing a nominal path planned by a motion planner.*

Let us view the nominal path as a sequence of straight line segments between “critical points”. Critical points are defined as points in the configuration space that the nominal path passes through. The points are chosen in a way that guarantees that between the critical points the robot may move in simple, straight line segments. The choice of critical points is dependent on the motion planner. Each motion planning algorithm is based on a different idea which in turn defines different critical points.

We use two algorithms for motion planning to illustrate our definitions: the visibility graph algorithm and a cell decomposition algorithm [9]. In the visibility graph algorithm, the critical points are the vertices of the obstacles. It is guaranteed that from each vertex one may see the next vertex that is on the path. Hence motion between the critical points is indeed on simple, straight line segments.

In the cell decomposition algorithm, we define the critical points as the mid-boundary points between the adjacent cells the nominal path passes through. It is guaranteed that the line segment between one point to

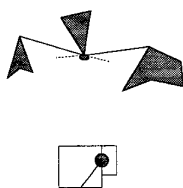


Figure 2: Critical regions in visibility graph and cell decomposition algorithms

the next is wholly contained in one cell. Hence it is guaranteed to be legal (i.e. not to collide with obstacles).

We now note the following observation. We may extend the critical points to critical regions which contain more points that satisfy the assumptions on which the motion planning algorithm is based. Hence, in the visibility graph algorithm, there are other points around the vertex from which both the previous and the next vertex are visible. In the cell decomposition algorithm, there are additional points near the mid-boundary point which satisfy the two “algorithmic” requirements. Firstly they will lead us from the current cell to the next cell, and secondly we may reach them from the current point by moving along a segment contained in the current cell. These extensions of critical points to critical regions is illustrated in Fig. 2.

The nominal path now clearly defines the accuracy requirements on localization, needed to guarantee completion of the path. Firstly, the robot should arrive with high probability in each of the critical regions. Secondly, in order to move between the critical regions, the robot should in principle just stick to the straight line segments connecting these regions. However, uncertainty in position may grow as long as the chances for collision with an obstacle stay low even with the uncertainty in position.

We stress the following point. The accuracy requirements are different in various regions of the configuration space, and for various nominal paths. For example, consider a path planned by a cell decomposition algorithm. In areas where the cells are large and have large boundaries between them, the critical regions are large. Hence the localization accuracy required is not very high. In other regions near obstacles, where cells are bound to be smaller, a higher localization accuracy may be required.

### 3 Varying Performance of Vision-Based Localization

We now turn to discuss the last factor involved - the performance of the sensor. In general the performance of the sensor varies across the configuration space and hence we can think of a map which associates with each configuration the accuracy of the localization result obtained by using the sensor at that configuration.

To illustrate this point, we will consider localization based on computation of camera motion based on several images. We assume that a camera 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 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. Details of this localization algorithm may be found in [3].

In [2] we have shown how to predict the uncertainty in the direction of translation resulting from this localization method. In the case of pure translation motion it is well known that the segments created by connecting pairs of corresponding points between the two images all lay on lines which meet at the Focus of Expansion (FOE). Thus the direction of translation (or FOE) may be found in principle by computing this intersection point from the given set of corresponding pairs. However, when the measured correspondences are noised, the lines on which the segments lay will not meet at one point. In this case we minimize an objective function which approximates a Maximum Likelihood Estimate for the FOE. See [2] for details.

Let us now consider two different positions with respect to the base image. Fig. 3 shows the segments created by corresponding pairs between the base image and the images at the two different positions. The + marks the location of the FOE in each case. It is intuitively clear that if the segments in Fig. 3(a) are perturbed a little, the FOE will move more than if the segments in part (b) of the figure will be perturbed. Thus the FOE computation is less stable in the first configuration than in the second.

To obtain a quantitative measure of the dispersion of the FOE due to noise in the corresponding points, we used the linear approximation method presented in [7]. That work develops a method for computing the covariance of an estimator which is computed by minimizing

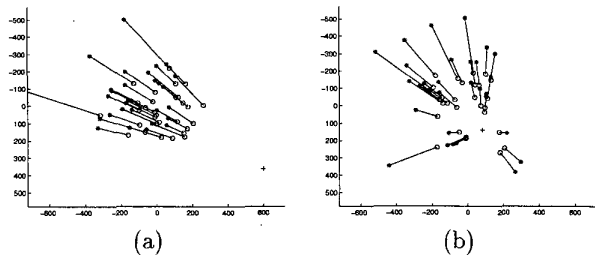


Figure 3: An example of (a) unstable and (b) stable configurations

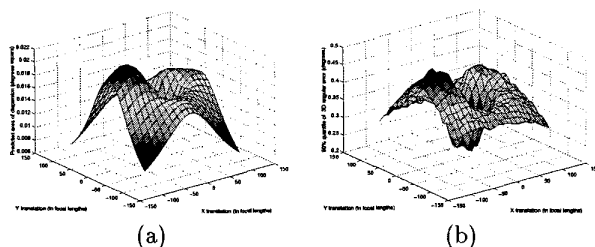


Figure 4: Uncertainty in direction of translation, as a function of position with respect to the base image. (a) Predicted uncertainty. (b) Empirical measure of dispersion. The graph shows the 90'th percent quantile of the 3D angular error distribution at each configuration.

an objective function. We applied that method to the objective function we use for computing the FOE.

As an example we refer to Fig. 4. Here the base image was taken from position  $(0, 0)$ . For each new position  $(X, Y)$ , we have estimated the error in determining the direction of translation to the new position. The theoretical prediction is shown in the left figure. In the right figure we show the empirical results. The 90'th percent quantile of the angular error in direction, is plotted as a function of position with respect to the base image. Clearly we can see that the empirical results agree with the predicted map.

As another example, consider the images in Fig. 5 taken in our lab. 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. Figure 6 shows the actual dispersion of 50 FOE estimates, and the dispersions described by the predicted and empirical covariance matrices. The an-



Figure 5: Base image (a) and image after forward translation (b)

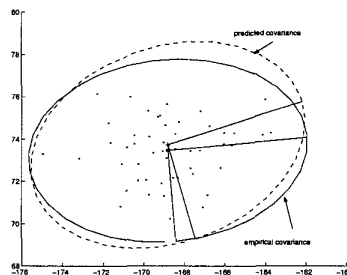


Figure 6: Predicted vs. empirical dispersion

gle between the principal axes of the two ellipses is 11.9 degrees. The ratios of the lengths of axes are 1.01 and 1.08. As is evident by these numbers and by looking at the figure, the prediction is quite accurate.

#### 4 Planning Updates Along the Route

We will now describe an example of using the active navigation paradigm and the performance map derived above for the vision sensor. Our robot uses the visibility graph method to plan paths in the work space. The planned path in this method passes through the vertices of the obstacles. As described previously, the accuracy requirement in this case defines critical regions around these vertices.

Consider now the position uncertainty but only in the direction which is orthogonal to the heading direction. Let us assign a single number to measure this uncertainty - for example 3 standard deviations. This uncertainty grows linearly according to our model. The distance from obstacles and the size of the critical regions define its maximum allowed value. This maximum allowed value changes along the route and is actually a piecewise constant function. We now compute the uncertainty in direction of translation with respect to two given base images, by the method described above. This allows us to infer the maximal uncertainty

in localization that will be obtained in the direction we are interested in. See Fig. 7.

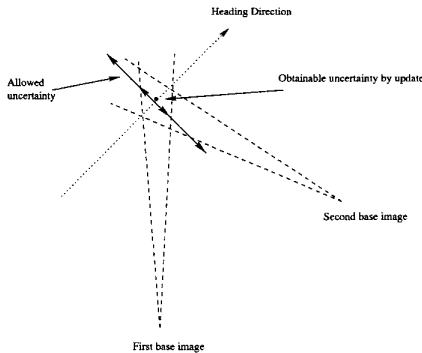


Figure 7: Allowed vs. obtainable uncertainty

We now let  $U_n(t)$  denote the maximal uncertainty allowed, and  $U_c(t)$  denote the uncertainty we will obtain by using the two base images. Let  $U(t)$  denote the linearly growing uncertainty in position. At all times, the uncertainty  $U(t)$  should be less than  $U_n(t)$ . Assume a position update at time  $t$  costs  $C(t)$ . The problem is to find times of update  $t_1, t_2, \dots$  which will ensure that

$$U(t) \leq U_n(t)$$

for  $t = 1, 2, \dots, T$ , and this with minimal cost of update operations. Fig. 8 illustrates this problem. This

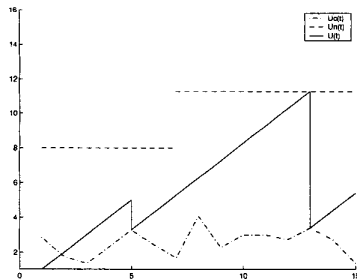


Figure 8: Find times for update actions which will ensure completion of the path with minimal cost (see text)

problem may be solved by the dynamic programming approach (see for example [4]). Let  $f(u; i, j)$  denote the cost of reaching from  $t = i$  to  $t = j$  with initial uncertainty  $u$  at time  $t = i$ . Let the uncertainty  $U(t)$  grow linearly with slope  $\alpha$ . If

$$u + \alpha(t - i) \leq U_n(t)$$

for all  $i \leq t \leq j$  then the no updates are necessary and  $f(u; i, j) = 0$ . Otherwise,  $f(u; i, j)$  may be computed recursively as

$$f(u; i, j) = \min_{i+1 \leq t \leq j-1} \{f(u; i, t) + C(t) + f(U_c(t); t, j)\}$$

Using the method of dynamic programming an efficient computational scheme may be employed to find the optimal times at which updates should be made.

## 5 Discussion

When considering vision as a sensor for mobile robot localization, one cannot expect to obtain the same level of performance uniformly over the work space. The active navigation paradigm is an approach which is designed to address this rather common characteristic of localization sensors. In this paper we have shown how we may use a vision sensor and the active navigation approach to plan sensing along the route, in a way which will allow a robot to successfully navigate along the route, while keeping the sensing cost to a minimum.

So far we have addressed the uncertainty in the direction of translation. There are additional issues which affect vision-based localization performance. An important example of such an issue is the variance in ability to find corresponding points between the images. We plan to address this question in future work.

## References

- [1] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Active navigation, 1999. In preparation.
- [2] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Computing the sensory uncertainty field of a vision based localization sensor. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 2000. To appear.
- [3] R. Basri, E. Rivlin, and I. Shimshoni. Visual homing: Surfing on the epipoles. *International Journal of Computer Vision*, 33(2), 1999.
- [4] R. E. Bellman and S. E. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, 1962.
- [5] J. Borenstein, H.R. Everett, L. Feng, and D. Wehe. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems*, 14(4):231-249, 1997.
- [6] 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.
- [7] R. M. Haralick. Propagating covariance in computer vision. In *Proc. of 12th ICPR*, pages 493-498, 1994.
- [8] 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.
- [9] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] S. Thrun. Finding landmarks for mobile robot navigation. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 958-963, 1998.
- [14] S. Thrun, W. Bugard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5:253-271, 1998.