Instituto de Sistemas e Robótica

Pólo de Lisboa

Visual Tracking for Mobile Robot Localization[1]

José Neira [2]

October 1996

RT-602-96

ISR-Torre Norte
Av. Rovisco Pais
1096 Lisboa CODEX
PORTUGAL

[1]This work has been financed by the HCM/ERNET programme, contract No. ERBCHRXCT 930381, of the European Commision.

[2]The author is with the Centro Politécnico Superior, Universidad de Zaragoza, Departamento de Informática e Ingeniería de Sistemas, c/María de Luna, 3, E-50015 Zaragoza, Spain (email : neira@prometeo.cps.unizar.es).

Abstract

In this report we analyse the fundamental aspects of the construction of indoor maps suitable for their use in navigation by a mobile robot. The limitations inherent to the use of *a priori* maps are highlighted. Fundamental problems related to the simultaneous construction of the map and the localization of the robot within this map using a monocular vision system are discussed. An experiment with the use of a monocular vision system to build a vertical edge based environmental map is proposed. For the construction of the map, the need to maintain information related to the correlations between the map features is shown. The correspondence problem, establishing a match between sensorial observations and map features, is solved by tracking the vertical edges in the sequence of images.

Contents

List of Figures

Chapter 1

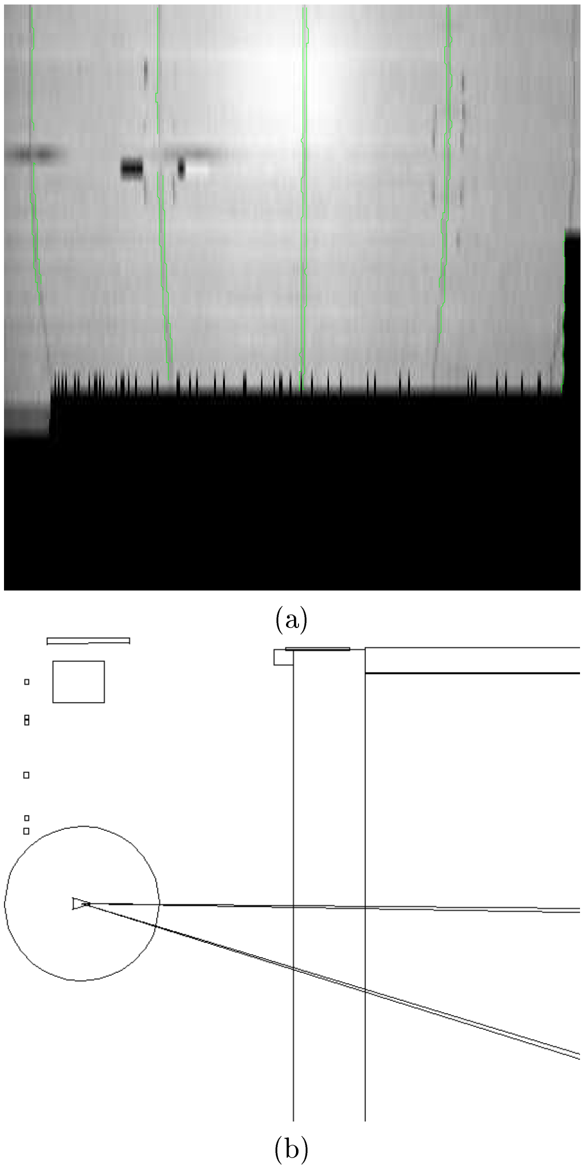
Introduction

Many robotic tasks, such as robot self-localization, require the use of a precise description of the robot environment. For this purpose, we usually provide the robot with an *a priori* map, obtained from the architectural designs of the building, or even by hand [8, 7, 6, 3]. The self-location process compares scene information obtained by some sensor or sensor combination with the scene description given by the *a priori* map and uses any discrepancy to correct errors inherent to dead-reckoning. This solution, although theoretically valid, proves unsatisfactory in some cases for the following reasons:

1. *Maps are incomplete.*
   It may prove too expensive to obtain a map with sufficient detail. This may result in the map not containing scene features that are detectable with sensors and thus potentially useful for tasks such as self-localization. Consider the situation shown in fig. 1.1. A mobile robot, the MACROBE [11], equipped with a monocular vision system, is shown in a location from which it detects clearly visible vertical edges corresponding to features of the wall in front of the robot. But since the *a priori* map contains only information regarding wall and corner location, these sensed edges cannot be used by the localization process.

2. *Maps are incorrect.*
   If *a priori* maps may lack information relevant to many tasks, in many cases they also include information that proves to be incorrect. This is very frequent in maps obtained from architectural designs, because they reflect the intention of the builders, but they may not reflect the final result. In fig. 1.2, a laser scan of the MACROBE robot shows that the actual wall location is not the one described in the map. In this case, sensor information that correctly reflects the scene features has to be discarded as spurious when compared with the incorrect map, but in general this can be a big source of errors if, as a result, the sensorial information is incorrectly interpreted. It is also frequent that initially correct maps become out-of-date as the scene changes due to many types of possible modifications (wall additions and eliminations, inclusion of large objects, etc.).

3. *Maps are imprecise.*
   It is difficult and costly to obtain a precise *a priori* map of the environment. In most cases, the precision with which the sensor is able to perceive the environment is higher



(a)

(b)

Figure 1.1: Vertical edges detected from the robot position (a), but not described in the map (b).

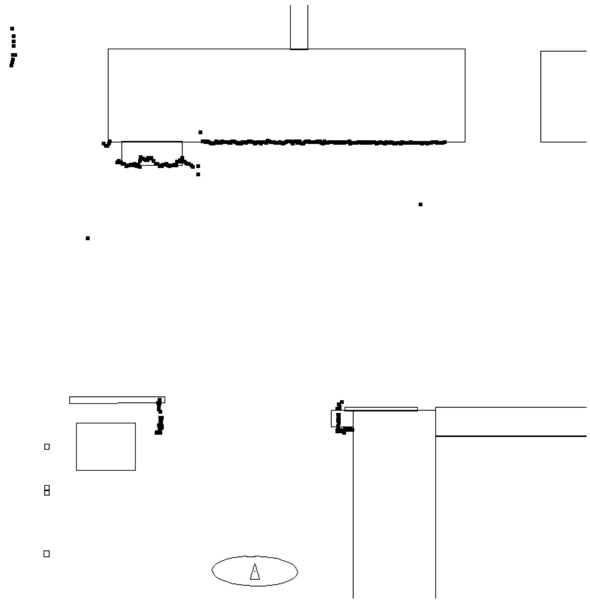


Figure 1.2: Laser scan showing that the location of some walls is not that which is described in the map.

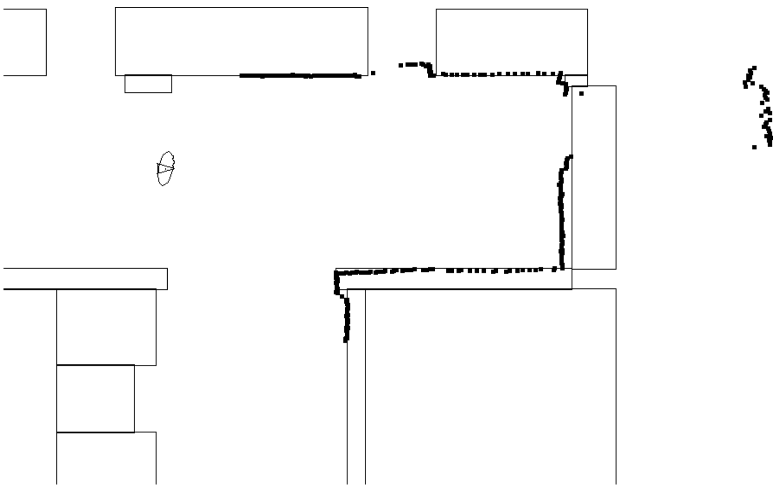


Figure 1.3: Some walls in the map are not described with sufficient precision for the estimation process to consider them adequately.

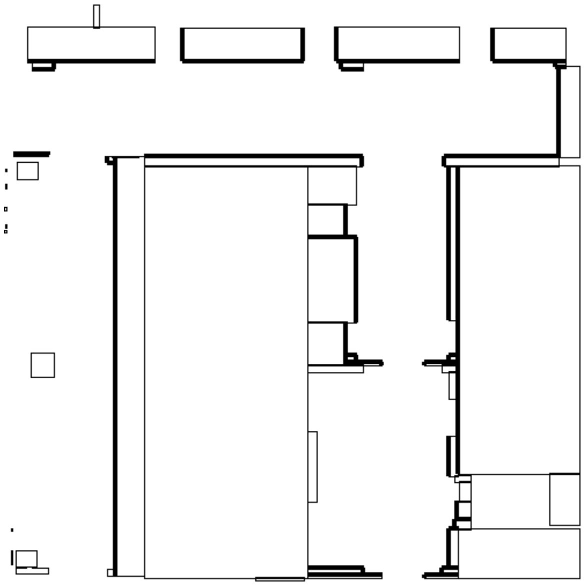


Figure 1.4: Highlighted segments corresponding to walls visible at some point of the robot trajectory.

than that of the map. This has two negative consequences: *first*, some potentially useful information may be considered spurious because the correspondence process is not able to match it with any feature of the map; *second*, even if some sensorial information is adequately matched in the map, the imprecision in the location of the corresponding features in the map will induce a systematic error in the estimation process. In fig. 1.3, the laser scan shows that the wall in front of the robot is not located exactly where the maps depicts it. This makes the correspondence process discard all the points corresponding to it as spurious. The scan also shows that the corridor is slightly less narrow than as it is described in the map. This causes the estimation process to be less precise than it would be possible.

4. *Maps contain unnecessary information.*
   Finding a match between a sensorial observation and its corresponding map feature is one of the most computationally expensive processes in mobile robot localization. This problem is aggravated if the map contains information that will never be perceived by the sensors. This happens mostly with maps obtained from architectural designs, that include occluded features. Fig. 1.4 highlights the map segments that were visible at some location of the robot path. As it can be seen, only a small percentage to the features have to be considered.

Given that sensors may perceive the robot environment more accurately, the evident alternative is to have the sensor *build* the map, rendering an always up-to-date description of the scene [9]. In this work, we investigate the limitations of *a priori* maps for mobile robot self-localization [5], and the fundamental issues related to automatic map building using as example a monocular vision system for environment perception [14].

This report is divided in three main parts: in chapter 2 we discuss the effect that imprecise maps may have in the self-location process. We also analyse the implications of the alternative map building process. Next, in chapter 3, we consider the fundamental problems of building an environmental map using monocular vision, and propose solutions that constitute a design for a map building experiment. In chapter 4 we draw the preliminary conclusions of this work.

This report also contains three substantial appendixes. We have tried to avoid mathematical detail throughout the discussion, and for this reason, we have concentrated it in the appendixes. These appendixes constitute a self-contained description of the mathematical tools used in this work, the SPmodel and SPfilter[13, 10].

Chapter 2

Map-based localization

Since robot motion errors are cumulative in nature, a mobile robot must periodically sense its environment to assess its location. For this purpose, the robot may have information about its environment, an *a priori* map, or alternatively it may *build* the map. In this chapter we analyse the advantages and drawbacks of each of these possibilities.

2.1 *A priori* maps

Consider providing the robot with a description of its environment so that the localization process can compare what it perceives with what it should perceive in order to correct any errors made by the motion mechanism. Let us study two cases:

- The *a priori* map is considered *perfect*. This is the first usual approach. Information given by the map is considered absolutely precise, and only the robot motion mechanism and the sensing processes are considered sources of possible errors. As we will see, this may have a considerable impact on the localization process.

- The *a priori* map is considered *imprecise*. Limitations found in the first approach lead to consider information given by the maps imprecise. This is done in the hope of avoiding problems related to perfect maps when they are not correct, but as we will see, the self-localization process becomes *pessimistic*, unnecessarily imprecise.

2.1.1 Perfect Maps

Consider the situation described in fig. 2.1. The mobile robot is instructed to perform the motion in a straight line as depicted. As odometry errors accumulate, the uncertainty in the position of the robot becomes larger (uncertainty in position is depicted as an ellipsoid; uncertainty in orientation is not depicted, although it is considered). In order to reduce this uncertainty, assume that the mobile robot uses a monocular vision system to acquire an image at each step of the path. This image is processed to extract vertical edges, corresponding to corners in the scene. Given that at each step there is only one image available, the sensor observations are projection rays between the camera optical center and the observed vertical edge. Note that the observations are drawn from the nominal robot location. Their discrepancy with the actual corner location is due both to robot positioning error and sensor measurement error.

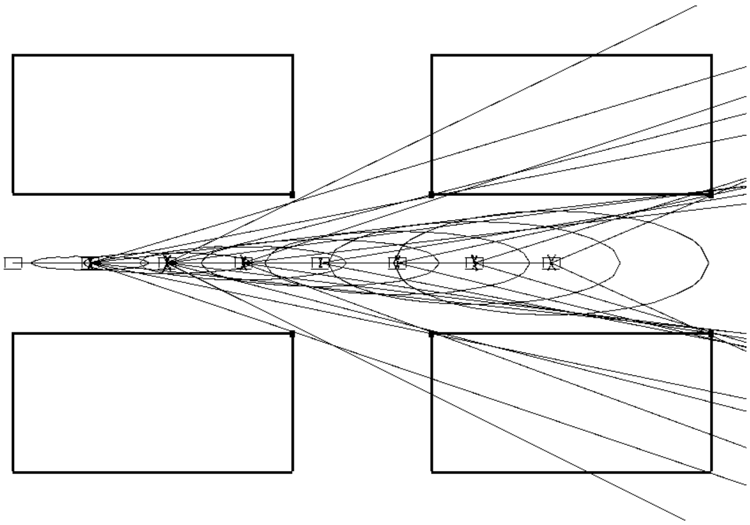


Figure 2.1: Robot nominal location and uncertainty along the path, and vertical edges observed from nominal location.

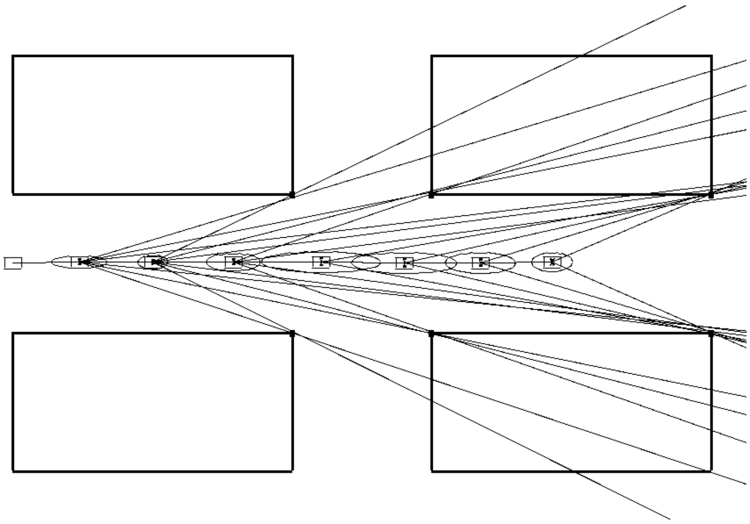
Assuming that correspondences are appropriately established (which constitutes a different but also crucial problem to adequately estimate the robot location and is studied in chapter 3), and that there are no spurious measurements (which is probably not the case in real situations, but also constitutes a problem that may be solved independently) the information may be fused (in our case with EIF, as described in section B.1.2) to estimate the real robot location and to reduce its uncertainty.

Fig. 2.2.a shows the result of the estimation process for a simulated robot motion and sensing process. Projection rays are drawn from the estimated robot location to show that the fusion process estimates a robot location from which the observations match better with their corresponding map features. Note that those observations that are nearer to the map corner fit better, which means that vertical edges corresponding to corners far away from the robot have less impact on the estimation than those that correspond to corners nearer to the robot. It is also interesting to see that the resulting estimation is less precise in the direction of motion. This is because a projection ray contributes basically with *angular* and *lateral* information, but with no *depth* information.

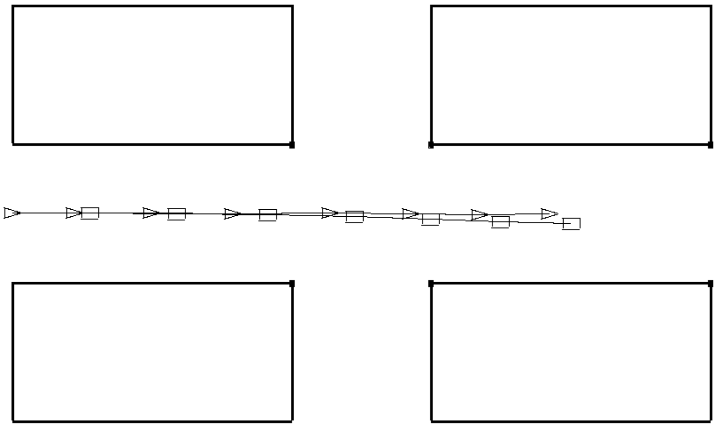
In this simulation, the robot tries to follow the nominal trajectory, so at each step the estimated location is used to compute the motion that should place the robot in the next nominal position. In fig. 2.2.b we compare the resulting estimated position with the odometric measurements of the robot. This illustrates the fact that, if the robot relied only on odometry, motion errors make its odometric measure drift from the real location in a considerable way.

If we use the estimated robot location to predict the location of the corners of the map which are visible (fig. 2.3), we will see that the predicted corner location is more accurate as the robot gets nearer to the corner, and becomes less precise when the robot is more distant to the predicted corner. Nevertheless, this prediction is never optimistic in the sense that the real corner location is always contained in the uncertainty ellipsoid of the prediction (it should be in at least 95% of the cases, as this is the certainty level used in the simulation).

Thus, as we may expect, if errors are only due to robot motion and sensor measurements,



(a)



(b)

Figure 2.2: Using a perfect map: (a) estimated robot location and its resulting uncertainty; observed projection rays from estimated position; (b) estimated robot location (triangle) .vs. robot location according to odometry (rectangle).

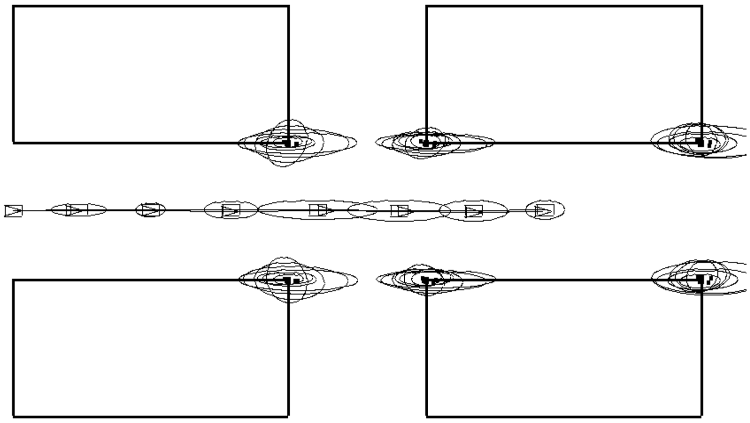


Figure 2.3: Predicted location of visible corners from the resulting estimated robot location.

then the assumption that the map is perfect allows us to precisely estimate the true location of the robot at each step of a trajectory and correct robot motion errors to follow the nominal path.

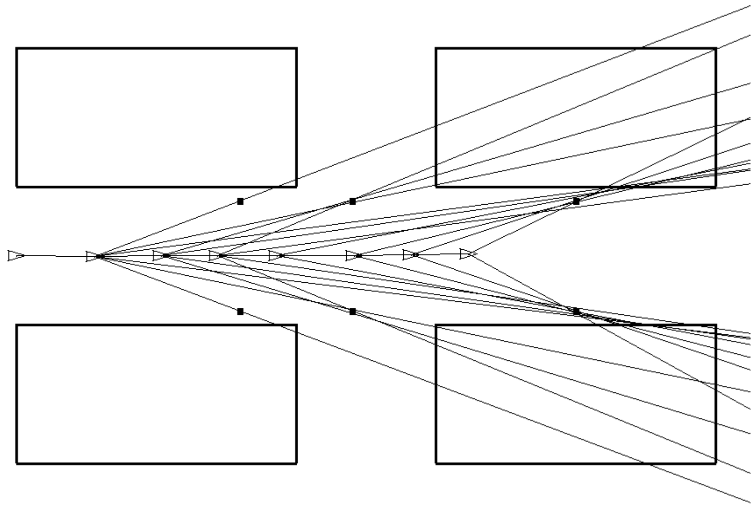
But what happens if, as it may normally be the case, the *a priori* map is not as accurate as we assume it is? As we said in the introduction, it is very expensive and difficult to have a really accurate map of the environment, so in many cases we have to settle for a less precise but affordable *a priori* map. How does our estimation process perform in this case? Consider the situation described in fig. 2.4.a. Assume that we have the same *a priori* map than in the preceding examples, but in the real scene the corners are located where the small squares in the figure show.

If we apply the same estimation process (section B.1.2), it will try to make the observations match the *a priori* corner locations as much as possible (fig. 2.4.b), making the estimated location of the robot *include* the discrepancy between the location of the corners according to the map and their real location. The consequence is that the resulting estimated robot location is inaccurate and optimistic (fig. 2.5). The real robot location is distant from its estimation, and it is not included in the uncertainty ellipsoid. The estimated location of the corners relative to the robot reference drifts from its *a priori* location towards its real location. Actually, the estimation process would consider that the observations do not match their paired features sufficiently well to consider the pair as being valid. This would make the system consider the observations spurious and, in that case, there could be no correction in the robot drift as the robot moves.

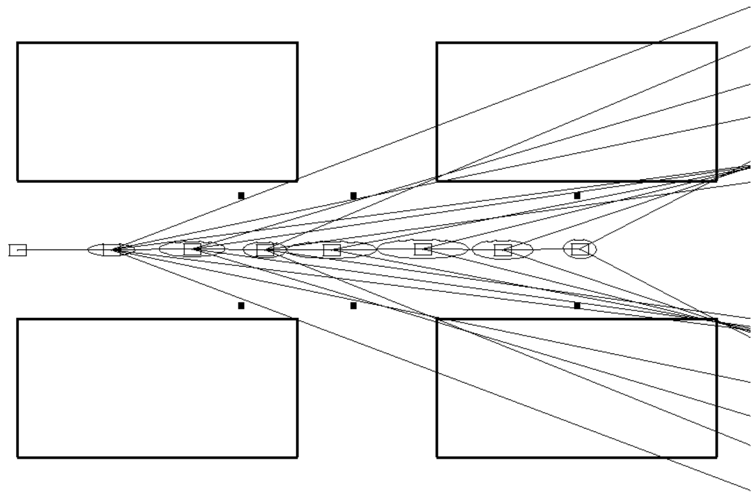
We can see that using an incorrect *a priori* map can make the robot self-localization process useless. In the following section we consider one possible alternative, that assumes imprecise *a priori* maps.

2.1.2 Imprecise Maps

The fundamental problem in the self-localization process described above is that the system considers perfect what is not perfect. One possible solution to allow the use of incorrect, or



(a)



(b)

Figure 2.4: Using an incorrect map: (a) real robot location and observed projection rays; (b) estimated robot location and observed projection rays according to estimation.

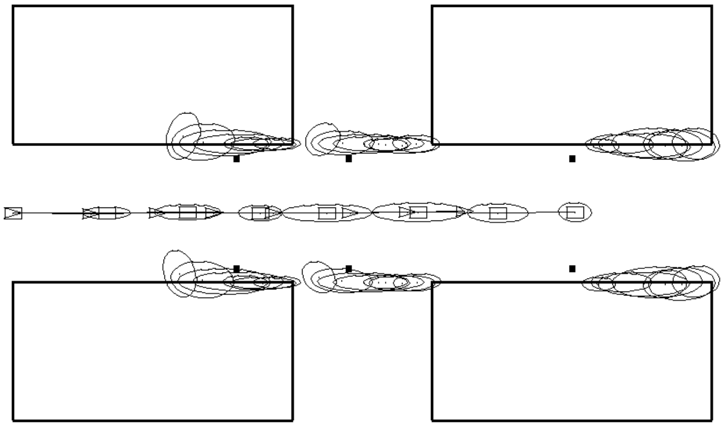


Figure 2.5: Discrepancy between estimated (small rectangle) and real robot location (small triangle); error in estimated corner location (relative to the robot reference).

rather imprecise, maps is evidently *not* to consider them perfect. That is, we may associate some uncertainty to the location of map features to reflect our imprecise knowledge about them. If fig. 2.6.a, the locations of visible corners of the scene have an associated uncertainty. Thus, in this case, there are three sources of uncertainty: the robot motion, the sensing process and the *a priori* map.

The self-localization process, described in section B.1.3, is applied in this case, and the result is shown in fig. 2.6.b. This procedure obtains an estimation of the robot location that is not optimistic, as in the former case, but given that the imprecision in the map is always considered the same, the estimated robot location is also imprecise. The problem lies in the fact that the procedure is considering an imprecise feature location, but it is only estimating the robot location, not the feature location, which is what it is actually observing.

This alternative is analyzed in the next section.

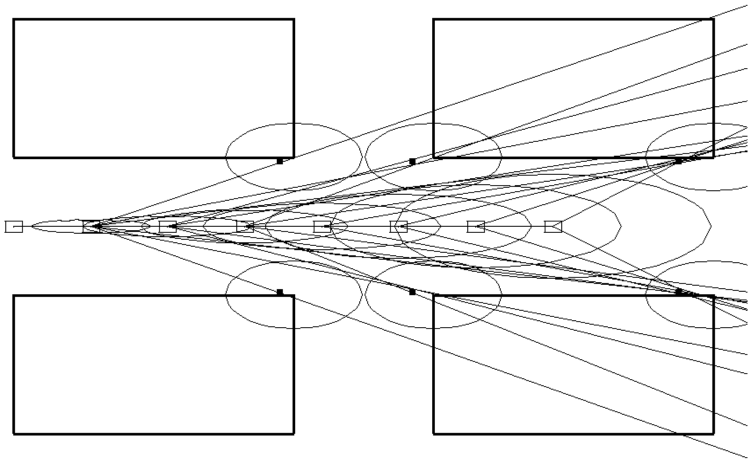
2.2 Constructing Maps

The cost and limitations inherent to the use of *a priori* maps, whether considered perfect or imprecise, lead us to the alternative of having the robot *build* a map of the environment where it navigates. Some of the potential advantages of building the map are:

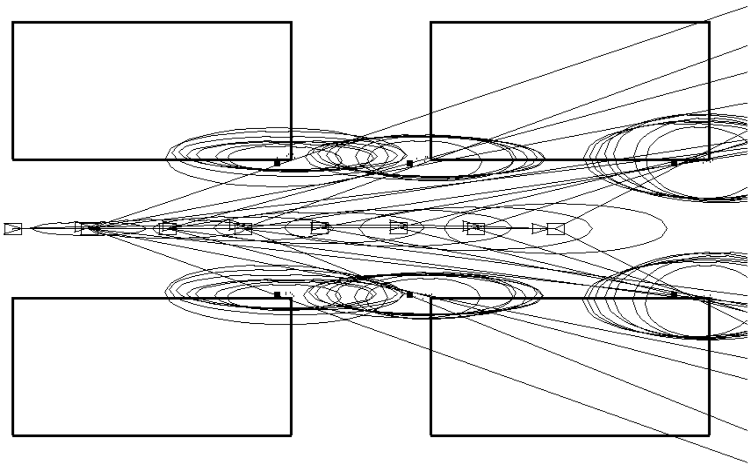
- Since the procedure is automatic, there is no cost in setting up the system. It should be able to navigate in any indoor environment.

- The map will be always up-to-date. When the indoor environment changes, the robot will acquire that information when navigating.

- The map will include the features that are perceivable by the sensor or sensor combination. There will not be any irrelevant information.



(a)



(b)

Figure 2.6: Using an imprecise map: (a) nominal robot location, observed projection rays and imprecise map corners; (b) estimated robot location (small rectangles), real robot location (small triangles) and predicted corner location according to estimation.

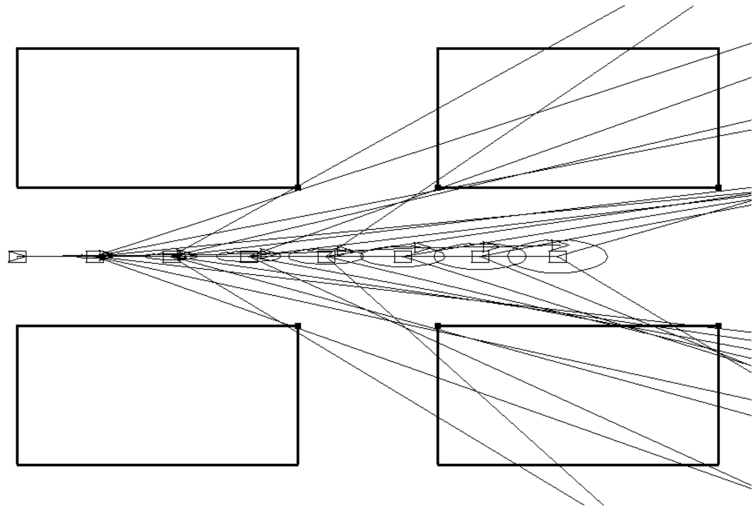


Figure 2.7: Map building: nominal robot location (small rectangle) and related uncertainty; real robot location (small triangle) and corner projection rays from nominal robot location.

- The map will be as precise and accurate as the sensors and robot used in building it. That means that the system will be used to its potential.

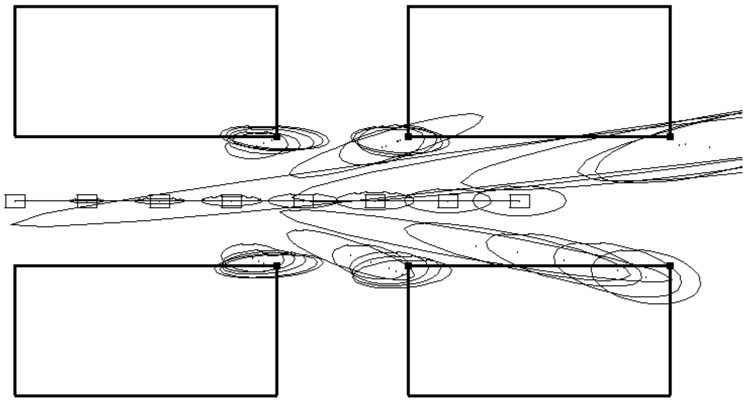
Map building can be limited to estimating the location of environment features, or additionally estimating the robot location with respect to these features. Each of these possibilities, which may serve a different purpose and has different advantages and problems, is analyzed next.

2.2.1 Map Building

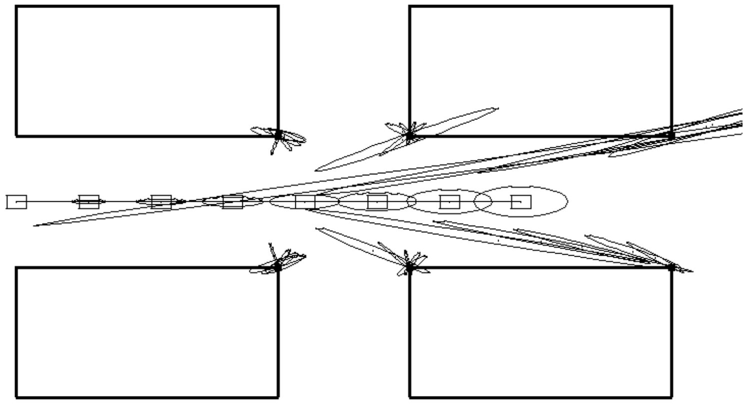
Some navigation tasks may require the robot to determine the precise location of the environment features. The question is whether it is also necessary to locate the robot with respect to the features. Consider the situation shown in fig. 2.7. In this simulated example, the robot observes the visible corners using a monocular vision system and tries to determine the location of visible corners. Since the estimated location of the robot is never corrected, the drift is clearly visible, and this has the consequence that the projection rays obtained from the monocular vision system also have an increasing error with respect to the observed features. Assuming that the correspondence problem is solved (this problem is analyzed in chapter 3), the system uses the first two non-parallel projection rays of each corner to determine the analytical corner location used as seed for the estimation process (estimation is done using an EIF described in section B.1.1).

The results of the estimation process are shown in fig. 2.8.a. The estimated corner location is rather poor, since the uncertainty in the robot location with respect to the world reference increases as the robot moves. We can also see that the initial corner location is very imprecise, specially for distant corners. This is because this initial location is obtained from projection rays that are nearly parallel in the case of distant corners, and for that reason the initial solution is very sensitive to sensor measurement errors.

In is important to see that, although the corner location with respect to the world reference



(a)



(b)

Figure 2.8: Estimated corner location: (a) with respect to the world reference, including uncertainty in robot position and sensor imprecision; (b) with respect to the robot reference, including only sensor measurement error.

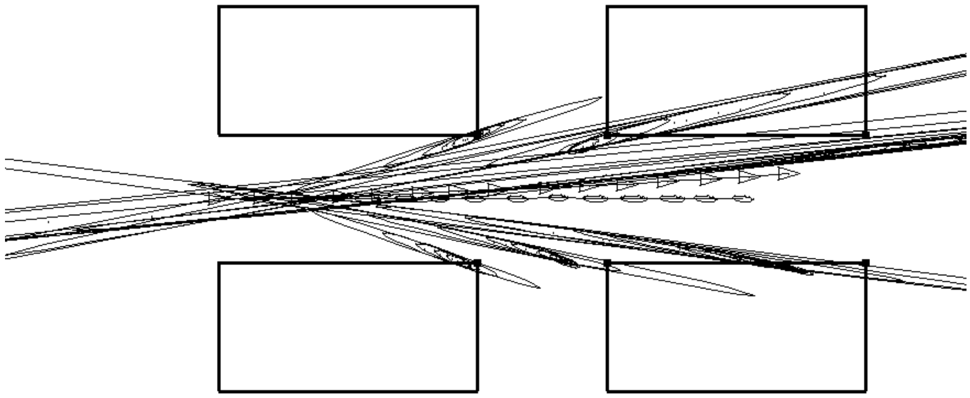


Figure 2.9: Map building *without* considering correlations: real robot location (small triangle) drifts from estimated robot location (small rectangle).

is rather poor and increasingly imprecise, its location *with respect to the robot* may be very accurate. That is, the map in the world reference is very imprecise, but in the robot reference it may be much more accurate (fig. 2.8.b). This means that if the mobile robot task requires the robot to precisely locate environment features from its location, then position assessment with respect to the features will be very precise, although robot position with respect to the world reference may be very imprecise.

2.2.2 Map Building and Self-localization

Some robot tasks require the system to precisely determine *both* the location of environment features *and* the robot position using sensor measurement of the environment. This confronts us with a rather difficult problem: our estimation mechanism, be it the EKF or the EIF, requires all integrated measurement to be *statistically independent*. In this case, we wish to use a sensor measurement to estimate both robot and feature location. Thus, these two estimations will become *statistically correlated*.

An environment map where correlations between all estimated features are maintained is usually referred to as a *stochastic map* [12]. If the amount of features is large, maintaining correlations is a computationally expensive process in time and space. From a theoretical point of view, statistical correlation will exist, but how *relevant* are these correlations[2]? Is it really necessary to maintain them?

To give insight into this matter, we consider first building a stochastic map maintaining *no* correlations, and try to determine whether the resulting stochastic map is accurate in the sense that its estimated feature locations are not *optimistic*.

For the simulated example used in this work we build the stochastic map as described in

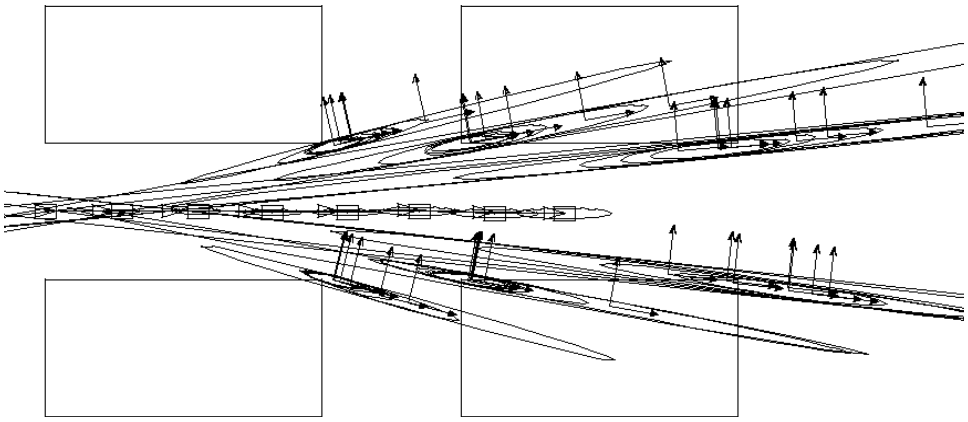


Figure 2.10: Map building *considering* correlations: real robot location (small triangle) does not drift from estimated robot location (small rectangle).

section B.2.1, without maintaining correlations between the map features (fig. 2.9). We can see that the resulting estimated robot location is inaccurate because the real robot location drifts out of the uncertainty ellipsoid. Given that the estimated corner location is influenced by this drift, this estimation is also inaccurate.

Thus, although costly, it is necessary to build and maintain the full stochastic map [5]. Fig. 2.10 shows the result of building the stochastic map and estimating the robot location considering correlations between features in the map, and between the robot and the map. In this case the real robot location is always within the estimated robot location, and thus there is no innacuracy in this sense. This is also the case for the features in the map. It is important to see that the location of features far from the robot will always be less precise than the location of those near the robot location. This suggests building the map relative to the robot reference instead of relative to a global reference. There exist the possibility that statistical correlation is reduced between features far from the robot, so that the stochastic map could be treated as a block-diagonal matrix. This possibility has to be verified experimentally.

Chapter 3

Map building using monocular vision

In this chapter we analyse the *correspondence problem* in map building. In general, this problem consists in matching sensorial information with environment model features that constitute the map. In our case, this is limited to determining the identity of the vertical edges observed in an image with respect to prior observations, as well as with respect to the built map. This is an important problem we have not considered up to this point. Success in solving it determines the correctness of the constructed map. When there is an *a priori* map available, this problem is not as complex, given that we always have the map as reference. But when we are building the map, specially in the initial steps, there is no reference to compare with, and this makes the correspondence problem error-prone.

In the following sections, we analyse some of the alternatives for solving the correspondence problem.

3.1 Coincidence as a basis for correspondence

In this particular case, the statistical tools for determining correspondences in monocular vision are of limited use. Consider the situation depicted in fig. 3.1. An observed monocular edge is drawn as a projection ray from the camera position. Due to the effect that the orientation uncertainty of the ray has on the position of points along it, more distant points along the ray are more uncertain (as shown). This is also true for the estimated location of, say corners, given the uncertain location of the robot: the position of more distant corners will be less precise. Thus, even if all corners are located at the same perpendicular distance to the projection ray, if we try to determine if the monocular edge correspond to any of the predicted corners, and we base our correspondence criterion solely on statistical distances (such as the mahalanobis distance, described in section A.3), we will favor correspondence of more distant corners (mahalanobis distance will be smaller), solely because their position is more uncertain. This contradictory situation should be avoided. Two possible solutions are:

- Instead of choosing the corner whose statistical distance is smaller, we should choose the corner whose *measured distance* (without considering imprecision) to the projection ray is smaller and whose statistical distance satisfies the hypothesis test to verify coincidence. This approach is of limited use because it may favour spurious matchings with

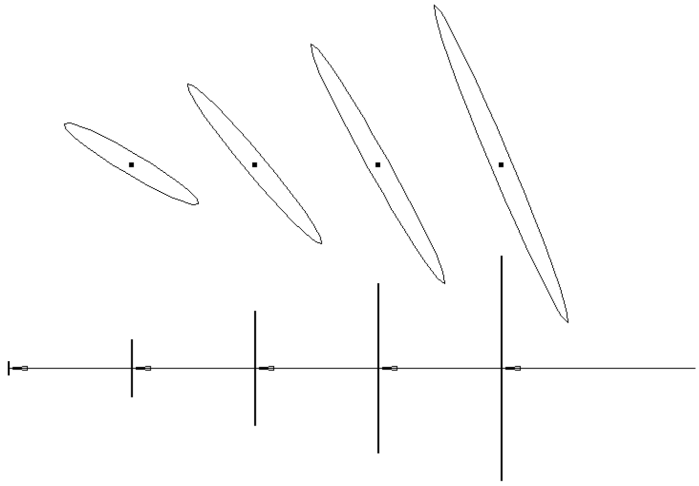


Figure 3.1: Statistical compatibility of a projection ray and a corner (a 2D point): more distant corners are more compatible.

corners that are located by chance nearer to the projection ray. This may be the case very frequent in long corridors, where there are many corners in the field of view.

- Choose the corner whose *distance to the camera* is smaller and whose statistical distance with the projection ray allows it to be considered coincident. This strategy would favour corners closer to the camera and certainly would limit the effect of distance in compatibility. In structured environments, the possibility of a spurious matching with features close to the camera is limited, *except* when all projection rays have a systematic error. The orientation error of the camera due to the robot position error is the same for all projection rays (fig. 3.2), and thus systematic for all observations. If we only applied this criterion, the two possible matchings of the figure would be spurious.

3.2 Local .vs. Global coherence

Instead of determining each matching separately, which limits coherence to a *local* one, we can decide some matching process that tries to maximize *global* coherence. That is, decide the matchings for *all* the projection rays of an image, so that they are *globally* coherent. This may be done in an iterative process by first accepting the most reliable pairings (projection rays with only one candidate corner), integrating the pairings to reduce the robot location error that is systematic in the projection rays of the image, and then deciding the rest of the matchings. This alternative can be implemented successfully [11], although it ignores the fact that the images are taken at small distances, and thus information related to the pairings obtained for the previous image may help decide the pairings for the next image. Another limitation of this approach is that it is not useful if there is no *a priori* information during the initial steps of the map building process.

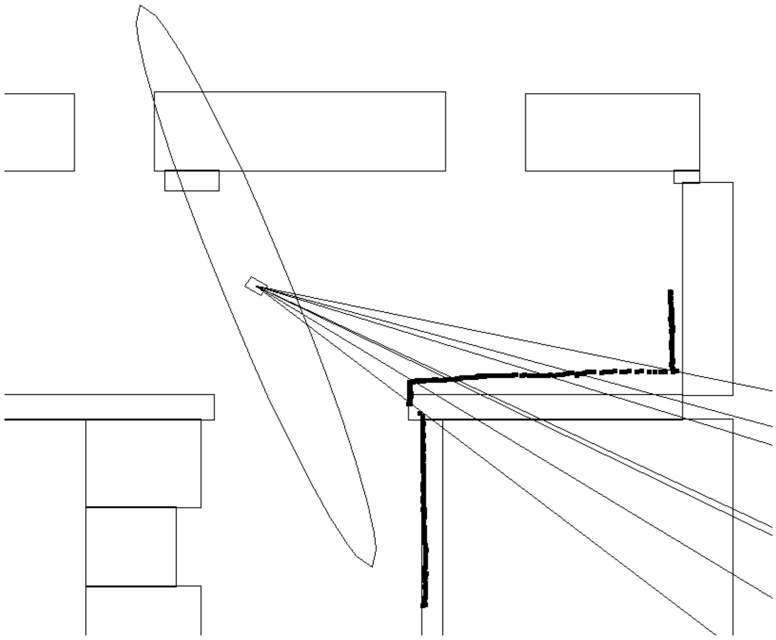


Figure 3.2: When all projection rays have a systematic error, the possibility of a spurious matching is higher. Counterclockwise the first projection ray would be incorrectly matched with the corner closest to it, which really corresponds to the second projection ray. Likewise, the last projection ray would be considered spurious, and its corresponding corner would be incorrectly matched with the second-to-last projection ray.

3.3 Tracking

Another alternative is to *track* the vertical edges in the sequence of images that the robot takes as it moves, which will give additional pairing information. This alternative may be an important source of information to reduce the complexity of the correspondence problem, because the robot trajectory is usually smooth, and so tracking can be successfully achieved.

We propose to use this last alternative. In the absence of an *a priori* map, it is the only alternative, and once an initial map is built, it may prove an important source of information for robust matching. The details of the tracking process are discussed in appendix C. Basically, the tracking algorithm should consider the following information:

- Image location of the middle point of the edge.
- Edge length. This parameter has to be used carefully since some long vertical edges will be partially visible on the image.
- Edge orientation in the image is *not* considered, since we are only dealing with vertical edges.
- Edge orientation according to gradient, although it may not be part of the state vector, may be used as an additional criterion to accept a matching.

It is necessary to point out that although tracking can be used with no *a priori* information, its robustness will largely depend on the type of motion that the robot undergoes from one image to the next. Straightforward motions of the robot will be handled with little problems by the tracking algorithm, but if a considerable change of orientation takes place, the tracking may completely loose the features. Actually, an interesting possibility is to make use of the fact that the relative location between one image and the next is not unknown, as the general structure and motion problem assumes. In fact, it is reasonable to assume that the robot motion between one image and the next will be sufficiently precise to *include* this information in the prediction phase of the estimation problem.

Chapter 4

Conclusions

In this work we studied the problem of map building and simultaneous robot self-localization using a monocular vision system. We have highlighted the fundamentallimitations of using *a priori* maps: the difficulty in having a precise *a priori* map available, and the problems that the use of imprecise maps cause to the process of estimating the robot location. These limitations lead to consider a map building procedure.

We have also discussed the most relevant aspects of the map building process. The two fundamental aspects of this process, and the solutions we propose can be summarized as follows:

- *The estimation problem.*
  If there is an *a priori* map available, the self-location is limited to estimating the robot location. But if the map has to be built at the same time, then self-location consists in estimating the location of the environment features, *and* estimating the robot location with respect to them. This makes the estimation process much more complex, since the same sensor measurements are used to estimate the map and the robot location, and thus the estimations are correlated. Disregarding correlations would make the estimation process much simpler, but the resulting map and estimated robot location become *optimistic*. Thus, the full *stochastic map* must be computed.

- *The correspondence problem.*
  The estimation process is based on the assumption that a correct correspondence between the sensor measurements and the environment features modeled in the map has been established. This makes correspondence a fundamental issue. The solution to this problem depends largely on the type of sensor being considered and also on the type of environment features that compose the map. Some sensors, such as laser, and environment features, such as walls, make correspondence a simpler problem to solve. In the case of laser, the mathematical tools used in determining if a laser point belongs to some wall are rather robust because indoor environments are structured enough to have very little confusion on whether a points belongs to a wall. This is not the case in the problem we study. We extract vertical edges for monocular vision images and try to establish a correspondence between the obtained projection rays and the features that constitute the map, wall corners. In this case, indoor environments, for example corridors, may have several corners visible and potentially compatible with a projection

ray of a single image. For this reason, we propose to perform *tracking* of the edges in the image as a way to help the correspondence process to establish matchings between the projection rays and the map corners.

This document is intended as a reference for the design of an experiment in robot navigation with a mobile robot for indoor environments and a monocular vision system for environment sensing. Nevertheless, other sensors can also be used for this purpose [2]. Given that the mathematical tools used here are adequate for multisensor systems, map building using some sensor combination, such as laser and vision, may result in very robust and reliable map building, the laser providing depth information while vision provides angular information.

Appendix A

Uncertain Geometry: the SPmodel

We use a representation model, the SPmodel, in which reference E is associated to the location of any type of geometric feature. The location of this reference with respect to a base reference is given by a transformation t_{WE} composed by two cartesian coordinates and an angle:

$$\mathbf{x}_{WE} = (x,\ y,\ \phi)^T \quad \text{where:} \quad t_{WE} = \text{Trans}(x,\ y) \cdot \text{Rot}(z,\ \phi)$$

The composition of two location vectors is denoted by $\oplus$, and the composition with the inverse is abbreviated by $\ominus$. Thus, given $\mathbf{x}_{AB} = (x_1,\ y_1,\ \phi_1)^T$ and $\mathbf{x}_{BC} = (x_2,\ y_2,\ \phi_2)^T$, their composition is calculated as follows:

$$\begin{aligned} \mathbf{x}_{AC} &= \mathbf{x}_{AB} \oplus \mathbf{x}_{BC} \\ &= (x_1 + x_2 \cos\phi_1 - y_2 \sin\phi_1, y_1 + x_2 \sin\phi_1 + y_2 \cos\phi_1, \phi_1 + \phi_2)^T \end{aligned} \tag{A.1}$$

Similarly,

$$\ominus\mathbf{x}_{AB} = (-x_1 \cos\phi_1 - y_1 \sin\phi_1, x_1 \sin\phi_1 - y_1 \cos\phi_1, -\phi_1)^T \tag{A.2}$$

A.1 Partiality: Binding Matrices

Different geometric features have different d.o.f. associated to their location. For example, the location of a robot in 2D is determined by three d.o.f., while the location of a point only by two. The d.o.f. that determine the location of a geometric entity are related to its *symmetries of continuous motion*. The symmetries of a geometric entity E are defined as the set $\mathcal{S}_E$ of transformations that preserve the element. For example, the symmetries of an infinite edge are the set of continuous translations (T_x) along the edge (fig. A.1). We represent the set of symmetries using a *row selection matrix* B_E, denominated *binding matrix of the feature*. In fig. A.1, the binding matrix for different types of geometric entities are given.

Binding matrices allow us to express and validate *coincidence*.

Definition A.1.1: Coincidence of Geometric Entities
Consider two geometric entities A and B of the same type (with associated binding matrices $B_A = B_B$), whose locations with respect to a base reference are given by:

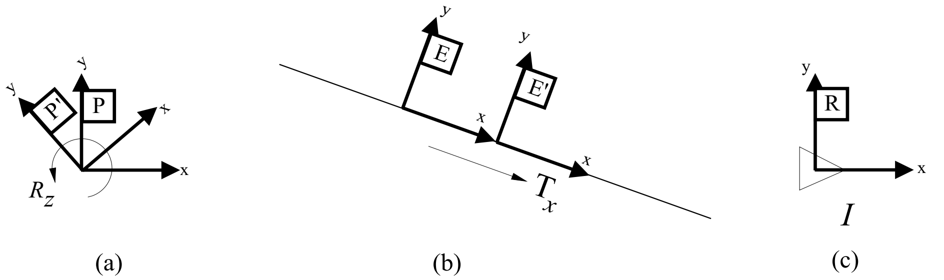


Figure A.1: Representation of (a) a 2D corner; (b) a 2D projection ray; (c) a robot in 2D.

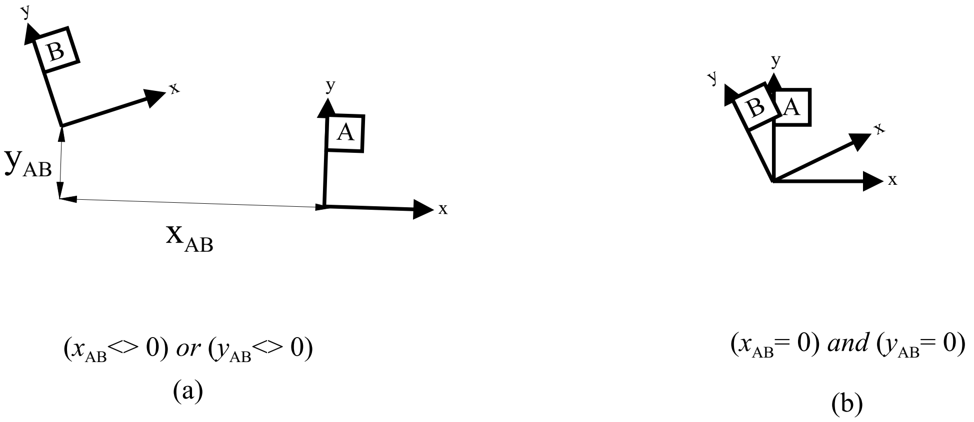


Figure A.2: Two points coincide if $x_{AB} = y_{AB} = 0$.

$$\mathbf{x}_{WA} = (x_A, y_A, \phi_A)^T \quad ; \quad \mathbf{x}_{WB} = (x_B, y_B, \phi_B)^T$$

According to eqs. (A.1) and (A.2), the relative location between A and B is given by:

$$\begin{aligned} \mathbf{x}_{AB} &= \ominus \mathbf{x}_{WA} \oplus \mathbf{x}_{WB} \\ &= ((y_B - y_A)\sin\phi_A + (x_B - x_A)\cos\phi_A, \\ &\quad (x_A - x_B)\sin\phi_A + (y_B - y_A)\cos\phi_A, \\ &\quad \phi_B - \phi_A)^T \\ &= (x_{AB}, y_{AB}, \phi_{AB})^T \end{aligned} \tag{A.3}$$

For the locations of A and B to coincide, the following must hold:

$$B_A \mathbf{x}_{AB} = B_B \mathbf{x}_{AB} = 0 \tag{A.4}$$

Example A.1.1: *Determining whether two points coincide*
Consider two points, whose location is represented by references A and B, respectively. The location of a point in 2D can be completely determined by the two first cartesian components of its location

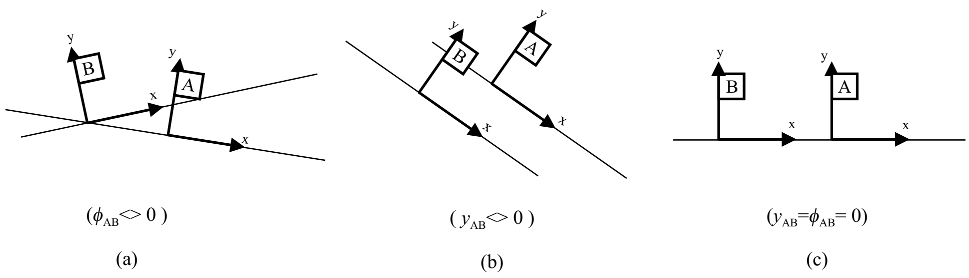


Figure A.3: Two edges coincide if $y_{AB} = \theta_{AB} = 0$.

vector. This is equivalent to saying that a 2D point has symmetries of rotation around a vector normal to the 2D plane, which corresponds to the third component of the location vector (fig. A.1.a). Given that this component contributes no location information, we have:

$$B_A = B_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Intuitively, the two points coincide when their distance $d = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$ equals zero (fig. A.2). This occurs when the two first components of their location vector are equal, i.e. $x_A = x_B$ and $y_A = y_B$. In the SPmodel, eq. (A.4) states that the two features coincide if:

$$B_A \mathbf{x}_{AB} = (x_{AB}, y_{AB})^T = 0$$

From eq. (A.3) we get:

$$\begin{aligned} (y_B - y_A) \sin \phi_A + (x_B - x_A) \cos \phi_A &= 0 \\ (x_A - x_B) \sin \phi_A + (y_B - y_A) \cos \phi_A &= 0 \end{aligned}$$

It is simple to verify that this equations are satisfied when $x_A = x_B$ and $y_A = y_B$. This result expresses the fact that the two points coincide if the relative position of their associated references is zero, *regardless of what the relative orientation of their associated reference is.* ◇

Definition A.1.2: Coincidence of Geometric Entities of diverse type
To express coincidence between different types of geometric elements, we use the binding matrix of a pairing. *In the case of two geometric entities of different type, whose location is represented by A and B respectively, one of the following equations expresses whether their locations coincide (up to symmetries):*

$$\begin{aligned} B_{AB} \mathbf{x}_{AB} &= 0 \qquad \textit{Direct Constraint} \\ B_{BA} \mathbf{x}_{BA} &= 0 \qquad \textit{Inverse Constraint} \end{aligned} \tag{A.5}$$

where B_{AB} and B_{BA} denote the binding matrices of the pairing.

Example A.1.2: *Determining whether a point belongs to an edge*
Consider a point and an edge, whose location is represented by P and E respectively. Given that the location of a 2D point is determined by two d.o.f. of position and the location of a 2D edge by one of position and one of orientation, and since we have chosen to associate the symmetries of the edge

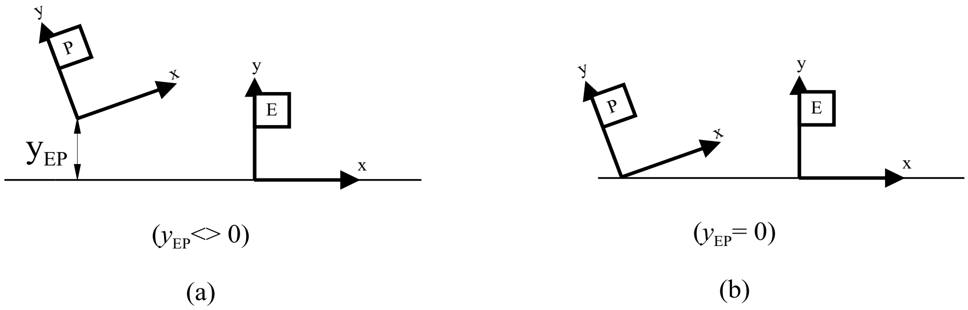


Figure A.4: A point belongs to an edge if $y_{EP} = 0$.

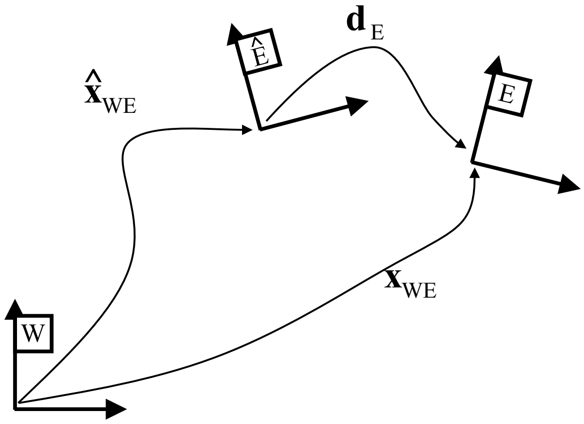


Figure A.5: Uncertain location of E in the SPmodel.

to its x axis, the d.o.f. that constraints reference P with respect to reference E corresponds to the y axis. Thus:

$$B_{EP} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

In this case eqs. (A.5) and gives:

$$B_{EP}\mathbf{x}_{EP} = y_{EP} = (x_E - x_P)\sin\phi_E + (y_P - y_E)\cos\phi_E = 0$$

This equation is only satisfied when $x_P = x_E + \frac{(y_P - y_E)\cos\phi_E}{\sin\phi_E}$, which means that the coordinates $(x_P, y_P)^T$ of the point must satisfy the equation of the edge of orientation ϕ_E that contains the point $(x_E, y_E)^T$, that is , the origin of the reference system of E.

◇

A.2 Imprecision: Perturbation Vectors

Since sensors give imprecise information, it is only possible to obtain an estimate of the location of a given geometric element. Most classical models of imprecision belong to one of two categories: *set-based* models and *probabilistic* models. For several reasons, we favor the use of probabilistic models [13]. In the SPmodel, the estimate of the location of a given entity E is denoted by $\hat{\mathbf{x}}_{WE}$, and the error associated to this estimate is expressed using a *differential location vector* $\mathbf{d}_E$, relative to the reference associated to the element, so that the true location of E is given by (fig. A.5):

$$\mathbf{x}_{WE} = \hat{\mathbf{x}}_{WE} \oplus \mathbf{d}_E \tag{A.6}$$

Since the d.o.f. of $\mathbf{d}_E$ corresponding to the symmetries of continuous motion contain no location information, we assign 0 to their corresponding values. We call *perturbation vector* a vector $\mathbf{p}_E$ formed by the non-null elements of $\mathbf{d}_E$. These two vectors are related by the binding matrix B_E:

$$\mathbf{d}_E = B_E^T \mathbf{p}_E \quad ; \quad \mathbf{p}_E = B_E \mathbf{d}_E$$

The information associated to the estimated location of a geometric element E is represented by a quadruple $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E, C_E, B_E)$, where:

$$\mathbf{x}_{WE} = \hat{\mathbf{x}}_{WE} \oplus B_E^T \mathbf{p}_E \; ; \; \hat{\mathbf{p}}_E = E[\mathbf{p}_E] \; ; \; C_E = Cov(\mathbf{p}_E)$$

We denominate this quadruple *uncertain location vector*. Note that the error associated to a location is expressed relative to the feature reference E and not to the base reference W. In this way the value of the covariance is not magnified by the distance of the feature to the base reference. This guarantees that the covariance values have a clear interpretation. The use of the binding matrix also makes the representation non-overparameterized.

A.3 Coincidence under Uncertainty

In this section we extend the concept of *coincidence* to uncertain geometric features and subfeatures.

Definition A.3.1: Coincidence for features
Given two uncertain geometric features of the same nature, whose location is given by $\mathbf{L}_{WA}$ and $\mathbf{L}_{WB}$ respectively, their locations coincide if the value of $B_B \hat{\mathbf{x}}_{AB}$ can be considered zero. The discrepancy between the locations of A and B is measured using the Mahalanobis distance:

$$D^2 = (B_A \hat{\mathbf{x}}_{AB})^T \left[B_A Cov(\mathbf{x}_{AB}) B_A^T \right]^{-1} B_A \hat{\mathbf{x}}_{AB} \tag{A.7}$$

where $\hat{\mathbf{x}}_{AB}$ and $Cov(\mathbf{x}_{AB})$ are obtained calculated as follows:

$$\begin{aligned} \hat{\mathbf{x}}_{AB} &= \ominus \hat{\mathbf{x}}_{WA} \oplus \hat{\mathbf{x}}_{WB} \\ Cov(\mathbf{x}_{AB}) &= J_{1\oplus}\{0, \hat{\mathbf{x}}_{AB}\} B_A^T C_A B_A J_{1\oplus}^T \{0, \hat{\mathbf{x}}_{AB}\} \\ &+ J_{2\oplus}\{\hat{\mathbf{x}}_{AB}, 0\} B_B^T C_B B_B J_{2\oplus}^T \{\hat{\mathbf{x}}_{AB}, 0\} \end{aligned} \tag{A.8}$$

Under the Gaussianity hypothesis, distance D^2 follows a chi-square distribution with $m =$ $\text{rank}(B_A)$ degrees of freedom. For a given significance level α, A and B can be considered coincident if:

$$D^2 \leq D^2_{m,\alpha} \tag{A.9}$$

Figure A.6: Verifying the location of a vertical edge with stereo vision.

Example A.3.1: *Verifying the location of a vertical edge with stereo vision*
Suppose a mobile robot with a stereo vision system is at some location from which the cameras detect a vertical edge at a location $\mathbf{L}_{RE}$ with respect to the robot reference. The system hypothesizes that it corresponds to an edge M in the environment map. Suppose that according to the map and estimated robot location, the predicted location of this edge would be $\mathbf{L}_{RM}$ (fig. A.6). In order to determine whether this hypothesis is valid, we must determine whether the location of E and M can be considered compatible. Let $\mathbf{L}_{RE}$ $(\hat{\mathbf{x}}_{RE}, \hat{\mathbf{p}}_E, C_E, B_E)$ and $\mathbf{L}_{RM} = (\hat{\mathbf{x}}_{RM}, \hat{\mathbf{p}}_M, C_M, B_M)$, where:

$$C_E = \begin{bmatrix} \sigma_{x_E}^2 & 0 \\ 0 & \sigma_{y_E}^2 \end{bmatrix} \quad ; \quad B_E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$
$$C_M = \begin{bmatrix} \sigma_{x_M}^2 & 0 \\ 0 & \sigma_{y_M}^2 \end{bmatrix} \quad ; \quad B_M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

For simplicity, let us have:

$$\hat{\mathbf{x}}_{EM} = (\hat{x}_{EM}, \hat{y}_{EM}, 0)^T$$

According to eq (A.7), the Mahalanobis distance between E and M would be:

$$D^2 = \frac{\hat{x}_{EM}^2}{\sigma_{x_E}^2 + \sigma_{x_M}^2} + \frac{\hat{y}_{EM}^2}{\sigma_{y_E}^2 + \sigma_{y_M}^2}$$

We can see that the discrepancy in the location of two points is directly proportional to the distance between the points (given by $\hat{x}_{EM}^2$ and $\hat{y}_{EM}^2$): the Mahalanobis distance increases with the distance between the points and thus the probability that the hypothesis will be accepted is reduced. Also, the Mahalanobis distance is inversely proportional to the covariance of the point locations (expressed by $\sigma_{x_E}^2$, $\sigma_{y_E}^2$, $\sigma_{x_M}^2$ and $\sigma_{y_M}^2$): the greater the uncertainty in the location of the features, the greater the probability that the hypothesis will be accepted. $\diamond$

Definition A.3.2: Coincidence for subfeatures

Let $\mathbf{L}_{WA}$ represent the uncertain location of a geometric feature, and let $\mathbf{L}_{WB}$ be a partial observation of the geometric feature represented by A. Assuming that the coincidence between A and B is described by the binding matrix B_{AB},, the location of A and B coincide if $B_{AB}\hat{\mathbf{x}}_{AB}$ can be considered zero. The discrepancy between the locations of A and B is again measured using the Mahalanobis distance:

$$D^2 = (B_{AB}\hat{\mathbf{x}}_{AB})^T \left[B_{AB} Cov(\mathbf{x}_{AB}) B_{AB}^T\right]^{-1} B_{AB}\hat{\mathbf{x}}_{AB}$$

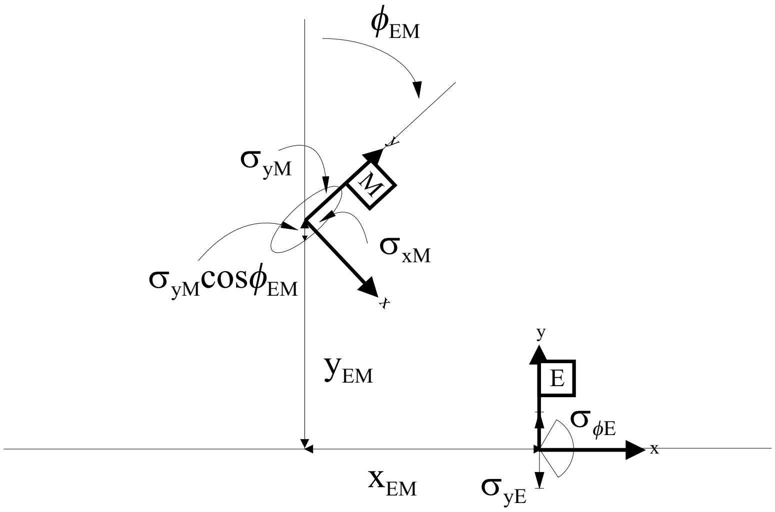


Figure A.7: Verifying the location of a vertical edge with monocular vision.

where $\hat{\mathbf{x}}_{AB}$ and $Cov(\mathbf{x}_{AB})$ are obtained from A.8. A hypothesis validation test similar to that of eq. (A.9) can be applied in this case.

Example A.3.2: *Verifying the location of a vertical edge with monocular vision*
Let us consider that the mobile robot of the preceding example is equipped with a monocular vision system with which it detects a vertical edge at a location $\mathbf{L}_{RE} = (\hat{\mathbf{x}}_{RE}, \hat{\mathbf{p}}_E, C_E, B_E)$. It then again hypothesizes that is corresponds to edge M in the map, whose predicted location with respect to the is $\mathbf{L}_{RM} = (\hat{\mathbf{x}}_{RM}, \hat{\mathbf{p}}_M, C_M, B_M)$ (fig. A.7). In this case we have:

$$C_E = \begin{bmatrix} \sigma^2_{y_E} & 0 \\ 0 & \sigma^2_{\phi_E} \end{bmatrix} \quad ; \quad B_E = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$C_M = \begin{bmatrix} \sigma^2_{x_M} & 0 \\ 0 & \sigma^2_{y_M} \end{bmatrix} \quad ; \quad B_M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Let:

$$\begin{aligned} \hat{\mathbf{x}}_{EM} &= \left(\hat{x}_{EM}, \hat{y}_{EM}, \hat{\phi}_{EM}\right)^T \\ B_{EM} &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \end{aligned}$$

In this case, according to eq. (A.7), the Mahalanobis distance between E and M would be:

$$D^2 = \frac{\hat{y}^2_{EM}}{\sigma^2_{y_E} + \hat{x}^2_{EM}\,\sigma^2_{\phi_E} + \sigma^2_{x_M}\,\sin^2\hat{\phi}_{EM} + \sigma^2_{y_M}\,\cos^2\hat{\phi}_{EM}}$$

The discrepancy is directly proportional to the *perpendicular* distance of the point to the edge (given by $\hat{y}^2_{EM}$): the more distant the point from the edge, the less likely that the hypothesis will accepted. This discrepancy is also inversely proportional to the covariance of this perpendicular distance, which depends on:

- the error associated to the position of E in the direction of the y axis (given by $\sigma^2_{y_E}$); the more imprecise the sensor, the more likely the hypothesis will be accepted;

- the error in the location of the closest point of edge E to point M, due to the orientation error of E (given by $\sigma^2_{\phi_E}$ and magnified by $\hat{x}^2_{EM}$); again the more imprecise the sensor, but also the more distant the position of M with respect to the robot, the more likely the hypothesis will be accepted;
- the error in the position of M in the direction of the y axis of E (given by $\sigma^2_{x_M} \sin^2 \hat{\phi}_{EM} + \sigma^2_{y_M} \cos^2 \hat{\phi}_{EM}$); the more uncertain the robot location, the more likely the hypothesis will be accepted.

◇

A.4 Operations with the SPmodel

A.4.1 Using differential location vectors

If $\mathbf{d}_A$ and $\mathbf{d}_B$ are differential location vectors then:

$$\begin{aligned} \mathbf{d}_A \oplus \mathbf{d}_B &\simeq \mathbf{d}_A + \mathbf{d}_B \\ \ominus \mathbf{d}_A &\simeq -\mathbf{d}_A \end{aligned}$$

The expressions to transform differential location vectors between references are:

$$\begin{array}{l} \mathbf{d}_A \oplus \mathbf{x}_{AB} = \mathbf{x}_{AB} \oplus \mathbf{d}_B \\ \mathbf{d}_B = J_{AB}^{-1}\, \mathbf{d}_A = J_{BA}\, \mathbf{d}_A \\ Cov(\mathbf{d}_B) = J_{BA}\, Cov(\mathbf{d}_A)\, J_{BA}^T \end{array}$$

where J_{AB} is the Jacobian corresponding to transformation $\mathbf{x}_{AB} = (x,\ y,\ \phi)^T$:

$$J_{AB} = \begin{bmatrix} \cos\phi & -\sin\phi & y \\ \sin\phi & \cos\phi & -x \\ 0 & 0 & 1 \end{bmatrix}$$

$$J_{BA} = J_{AB}^{-1} = \begin{bmatrix} \cos\phi & \sin\phi & x\sin\phi - y\cos\phi \\ -\sin\phi & \cos\phi & x\cos\phi + y\sin\phi \\ 0 & 0 & 1 \end{bmatrix}$$

A.4.2 Centering an Uncertain Location

Given an uncertain location $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E, C_E, B_E)$, where $\hat{\mathbf{p}}_E \neq 0$, it can be transformed to $\mathbf{L}_{WE'} = (\hat{\mathbf{x}}_{WE'}, \hat{\mathbf{p}}_{E'}, C_{E'}, B_E)$, where $\hat{\mathbf{p}}_{E'} = 0$ (fig. A.8.a) in the following way:

$$\begin{aligned} \hat{\mathbf{x}}_{WE'} &= \hat{\mathbf{x}}_{WE} \oplus B_E^T \hat{\mathbf{p}}_E \\ C_{E'} &= \left(B_E J_{2\oplus}^{-1} \left\{ B_E^T \hat{\mathbf{p}}_E, 0 \right\} B_E^T \right) C_E \left(B_E J_{2\oplus}^{-1} \left\{ B_E^T \hat{\mathbf{p}}_E, 0 \right\} B_E^T \right)^T \\ B_{E'} &= B_E \end{aligned}$$

where $J_{2\oplus}$ is the Jacobian of the composition of two location vectors:

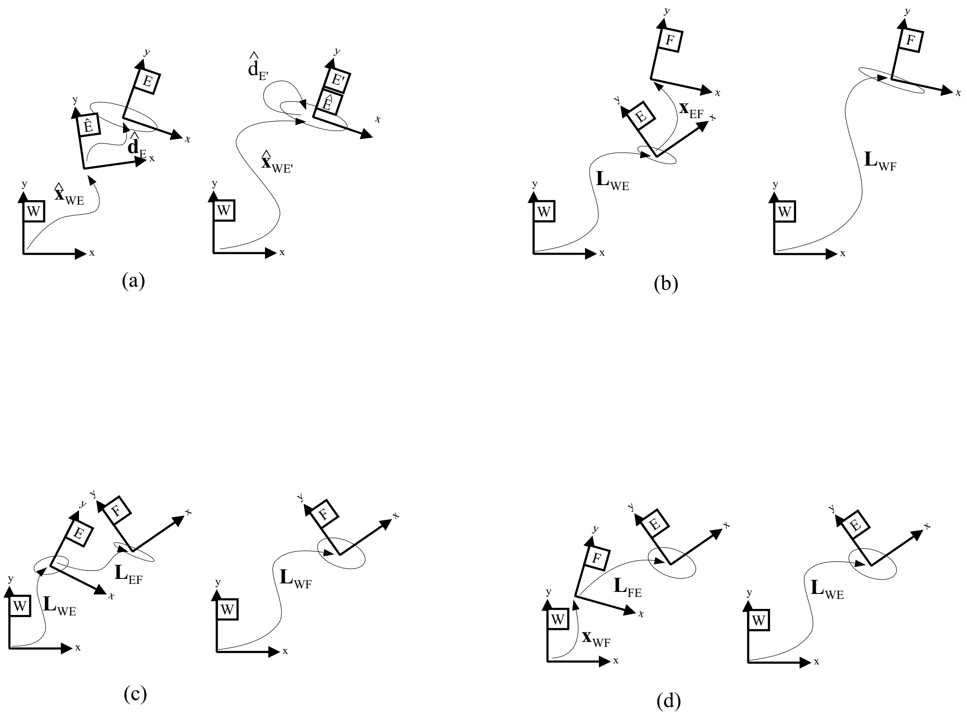


Figure A.8: Operations with the SPmodel: (a) centering an uncertain location; (b) changing the associated reference of an uncertain location; (c) composing two uncertain locations; (d) changing the base reference.

$$J_{1\oplus}\{\mathbf{x}_1, \mathbf{x}_2\} = \left.\frac{\partial(\mathbf{y} \oplus \mathbf{z})}{\partial \mathbf{y}}\right|_{\mathbf{y}=\mathbf{x}_1,\, \mathbf{z}=\mathbf{x}_2} = \begin{bmatrix} 1 & 0 & -x_2 \sin\phi_1 - y_2 \cos\phi_1 \\ 0 & 1 & x_2 \cos\phi_1 - y_2 \sin\phi_1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$J_{1\oplus}\{0, \mathbf{x}\} = \begin{bmatrix} 1 & 0 & -y \\ 0 & 1 & x \\ 0 & 0 & 1 \end{bmatrix}$$

$$J_{2\oplus}\{\mathbf{x}_1, \mathbf{x}_2\} = \left.\frac{\partial(\mathbf{y} \oplus \mathbf{z})}{\partial \mathbf{z}}\right|_{\mathbf{y}=\mathbf{x}_1,\, \mathbf{z}=\mathbf{x}_2} = \begin{bmatrix} \cos\phi_1 & -\sin\phi_1 & 0 \\ \sin\phi_1 & \cos\phi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$J_{2\oplus}\{\mathbf{x}, 0\} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

From here on, all uncertain location will be assumed centered. This operation considerably simplifies the expressions and calculations involved in fusing different observations.

A.4.3 Calculating the Inverse of an Uncertain Location: $\mathbf{L}_{BA} = \ominus\mathbf{L}_{AB}$

Given an uncertain location $\mathbf{L}_{AB} = (\hat{\mathbf{x}}_{AB}, \hat{\mathbf{p}}_B, C_B, B_B)$, its inverse $\mathbf{L}_{BA} = (\hat{\mathbf{x}}_{BA}, \hat{\mathbf{p}}_A, C_A, B_A)$ can be calculated in the following way:

$$
\begin{aligned}
\mathbf{x}_{BA} &= \ominus \mathbf{x}_{AB} \\
&= \ominus \left(\hat{\mathbf{x}}_{AB} \oplus B_B^T \mathbf{p}_B \right) \\
&= \ominus B_B^T \mathbf{p}_B \ominus \hat{\mathbf{x}}_{AB} \\
&= \ominus B_B^T \mathbf{p}_B \oplus \hat{\mathbf{x}}_{BA} \\
&= \hat{\mathbf{x}}_{BA} \ominus J_{AB} B_B^T \mathbf{p}_B \\
&\simeq \hat{\mathbf{x}}_{BA} \oplus -J_{AB} B_B^T \mathbf{p}_B
\end{aligned}
$$

Therefore:

$$
\begin{aligned}
\mathbf{p}_A &= -B_B J_{AB} B_B^T \mathbf{p}_B \\
B_A &= B_B \\
C_A &= B_B J_{AB} B_B^T C_B B_B J_{AB}^T B_B^T
\end{aligned}
$$

A.4.4 Changing the Associated Reference: $\mathbf{L}_{WF} = \mathbf{L}_{WE} \oplus \mathbf{x}_{EF}$

Given the estimated location of a geometric element E, represented by $\mathbf{L}_{WE}$, we may choose to represent it by an alternative reference F. If $\mathbf{x}_{EF}$ represents the relative location between E and F, we have:

$$
\begin{aligned}
\mathbf{x}_{WF} &= \mathbf{x}_{WE} \oplus \mathbf{x}_{EF} \\
&= \hat{\mathbf{x}}_{WE} \oplus B_E^T \mathbf{p}_E \oplus \mathbf{x}_{EF} \\
&= \hat{\mathbf{x}}_{WE} \oplus \mathbf{x}_{EF} \oplus J_{FE} B_E^T \mathbf{p}_E
\end{aligned}
$$

Thus the estimated location of F with respect to W will be given by $\mathbf{L}_{WF} = (\hat{\mathbf{x}}_{WF}, \hat{\mathbf{p}}_F, C_F, B_F)$, where:

$$
\begin{aligned}
\hat{\mathbf{x}}_{WF} &= \hat{\mathbf{x}}_{WE} \oplus \mathbf{x}_{EF} \\
\hat{\mathbf{p}}_F &= B_E \, J_{FE} \, B_E^T \, \hat{\mathbf{p}}_E \\
C_F &= B_E J_{FE} B_E^T \; C_E \; B_E J_{FE}^T B_E^T \\
B_F &= B_E
\end{aligned}
$$

This operation is useful for aligning the reference of a geometric element to estimate its geometric relations with another geometric element (fig. A.8.b).

A.4.5 Composing two Uncertain Locations: $\mathbf{L}_{WE} = \mathbf{L}_{WF} \oplus \mathbf{L}_{FE}$

Consider a geometric element E, whose uncertain location with respect to reference F is expressed by $\mathbf{L}_{FE}$. Suppose this reference is also affected by a location error with respect to W, expressed by $\mathbf{L}_{WF}$. If this error is *independent* from the error which affects E, then the uncertain location of E with respect to W is obtained as follows: location vector $\mathbf{x}_{WE}$ will be given by:

$$
\begin{aligned}
\mathbf{x}_{WE} &= \mathbf{x}_{WF} \oplus \mathbf{x}_{FE} \\
&= \hat{\mathbf{x}}_{WF} \oplus B_F^T \mathbf{p}_F \oplus \hat{\mathbf{x}}_{FE} \oplus B_E^T \mathbf{p}_E \\
&= (\hat{\mathbf{x}}_{WF} \oplus \hat{\mathbf{x}}_{FE}) \oplus (J_{EF}\, B_F^T \mathbf{p}_F \oplus B_E^T \mathbf{p}_E) \\
&\simeq (\hat{\mathbf{x}}_{WF} \oplus \hat{\mathbf{x}}_{FE}) \oplus J_{EF}\, B_F^T \mathbf{p}_F + B_E^T \mathbf{p}_E
\end{aligned}
$$

Thus, the uncertain location of E with respect to W is given by $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E^W, C_E^W, B_E)$, where:

$$
\begin{aligned}
\hat{\mathbf{x}}_{WE} &= \hat{\mathbf{x}}_{WF} \oplus \hat{\mathbf{x}}_{FE} \\
\mathbf{p}_E^W &= B_E J_{EF}\, B_F^T \mathbf{p}_F + \mathbf{p}_E \\
C_E^W &= B_E J_{EF} B_F^T C_F B_F J_{EF}^T B_E^T + C_E
\end{aligned}
$$

This operation can be useful in situations where a sensor mounted on the mobile robot obtains the estimated location of a feature. It allows us to obtain the estimated location of the feature with respect to the base reference, taking into account both the sensor error and the robot positioning error (fig. A.8.c).

A.4.6 Changing the Base Reference: $\mathbf{L}_{WE} = \mathbf{x}_{WF} \oplus \mathbf{L}_{FE}$

Given a geometric element E, whose uncertain location with respect to a reference F is given by $\mathbf{L}_{FE}$, we may choose to represent it with respect to another base reference, say W. If $\mathbf{x}_{WF}$ represents the relative location between the bases references, we have:

$$
\begin{aligned}
\mathbf{x}_{WE} &= \mathbf{x}_{WF} \oplus \mathbf{x}_{FE} \\
&= \mathbf{x}_{WF} \oplus \hat{\mathbf{x}}_{FE} \oplus B_E^T \mathbf{p}_E
\end{aligned}
$$

Thus, the uncertain location of E with respect to W will be given by $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E, C_E, B_E)$, where:

$$
\hat{\mathbf{x}}_{WE} = \mathbf{x}_{WF} \oplus \hat{\mathbf{x}}_{FE}
$$

This operation can be useful when a sensor mounted on the mobile robot estimates the location of a feature with respect to the reference associated to the sensor. Using the relative location between the sensor and the robot, we can determine the uncertain location of the feature with respect to the reference of the robot (fig. A.8.d).

Appendix B

Multisensor Fusion

In [13], the SPmodel is used to establish a general integration mechanism that allows to obtain a *suboptimal estimation of location* for objects or features from a set of partial and uncertain sensorial observations. The estimation of the location of an object or feature from a set of geometric observations is nonlinear, due to the existence of orientation terms, and can be solved using the *extended Kalman filter* (EKF) or the *extended Information filter* (EIF) [4, 1]. In this report we use the 2D version of the SPmodel along with the specialized versions of the EKF and EIF for the SPmodel, to estimate the location of a mobile robot from a set of partial and imprecise observations of features in the robot's environment.

B.1 The Extended Information Filter

The extended Information filter can be stated as follows: let $\mathbf{x}$ be the state vector whose value is to be estimated, and let there be n independent and possibly partial observations $\mathbf{y}_k$ of $\mathbf{x}$, where $k \in \{1, \ldots, n\}$, affected by white Gaussian noise:

$$\hat{\mathbf{y}}_k = \mathbf{y}_k + \mathbf{u}_k \;\; ; \;\; \mathbf{u}_k \sim N(0, S_k)$$

Let each observation $\mathbf{y}_k$ be related to $\mathbf{x}$ by an implicit nonlinear function of the form $\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) = 0$. We use a first order approximation of $\mathbf{f}_k$:

$$\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) \simeq \mathbf{h}_k + H_k(\mathbf{x} - \hat{\mathbf{x}}) + G_k(\mathbf{y}_k - \hat{\mathbf{y}}_k)$$

where:

$$\mathbf{h}_k = \mathbf{f}_k(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k) \;\; ; \;\; H_k = \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k)} \;\; ; \;\; G_k = \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{y}}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k)} \tag{B.1}$$

The estimate $\hat{\mathbf{x}}_n$ of the state vector and its covariance P_n after integrating the n measurements are:

$$\hat{\mathbf{x}}_n = P_n M_n \;\; ; \;\; P_n^{-1} = Q_n = \sum_{k=1}^{n} F_k \;\; ; \;\; M_n = -\sum_{k=1}^{n} N_k \tag{B.2}$$

where:

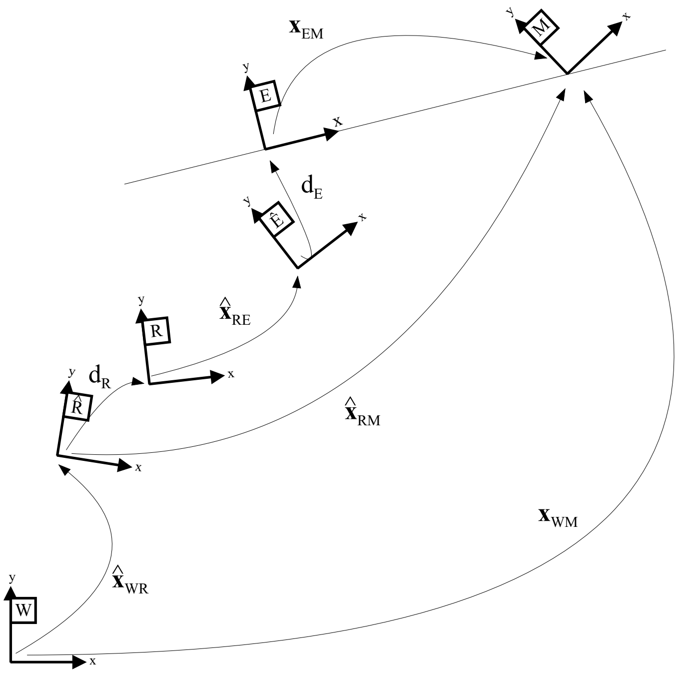


Figure B.1: References involved in the integration of a monocular projection ray to the estimation of a corner.

$$F_k = H_k^T (G_k S_k G_k^T)^{-1} H_k \ \ ; \ \ N_k = H_k^T (G_k S_k G_k^T)^{-1} \mathbf{h}_k \tag{B.3}$$

This is the non-recursive formulation of the information filter, which is equivalent to a least squares estimation in batch mode: integrating a block of n measurements at the same time. Formulations of the recursive information filter and Kalman filter can be found in [13]. This integration mechanism can be applied to a wide variety of integration problems. The cases of interest for the present work are given below.

B.1.1 Estimating the location of a corner with monocular vision

Let $\mathbf{L}_{RE} = (\hat{\mathbf{x}}_{RE}, \hat{\mathbf{p}}_E, C_E, B_E)$ be the estimated location of a corner E (a 2D point) with respect to the robot reference R. Let $\mathbf{L}_{RP} = (\hat{\mathbf{x}}_{RP}, \hat{\mathbf{p}}_P, C_P, B_P)$ be the observed projection P of the location of the corner (a 2D point) with respect to the robot reference R, obtained from a monocular vision system. Hypothesizing that the observation P belongs to the element E, we can use it to improve the estimate of the location of the element. In this case, the state to be estimated is represented by the perturbation vector of E, and the measurement by the perturbation vector of P:

$$\begin{aligned} \mathbf{x} &= \mathbf{p}_E \\ \mathbf{y}_k &= \mathbf{p}_P \end{aligned}$$

Coincidence between E and P can be described by means of the *inverse constraint*. Thus, we have (fig. B.1):

$$
\begin{aligned}
\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) &= B_{PE}\mathbf{x}_{PE} \\
&= B_{PE} \ominus \mathbf{x}_{RP} \oplus \mathbf{x}_{RE} \\
&= B_{PE} \ominus (\hat{\mathbf{x}}_{RP} \oplus \mathbf{d}_P) \oplus \hat{\mathbf{x}}_{RE} \oplus \mathbf{d}_E \\
&= B_{PE} \ominus \mathbf{d}_P \ominus \hat{\mathbf{x}}_{RP} \oplus \hat{\mathbf{x}}_{RE} \oplus \mathbf{d}_E \\
&= B_{PE} \ominus \mathbf{d}_P \oplus \hat{\mathbf{x}}_{PE} \oplus \mathbf{d}_E \\
&= B_{PE} \ominus B_P^T \mathbf{p}_P \oplus \hat{\mathbf{x}}_{PE} \oplus B_E^T \mathbf{p}_E \\
&= B_{PE}(\ominus B_P^T \mathbf{y}_k \oplus \hat{\mathbf{x}}_{EP} \oplus B_E^T \mathbf{x}) \\
&= 0
\end{aligned}
$$

In this case we have:

$$
B_E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \; ; \; B_P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \; ; \; B_{PE} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}
$$

If we consider both estimations centered, we have:

$$
\begin{aligned}
\mathbf{h}_k &= B_{PE}\hat{\mathbf{x}}_{PE} \\
H_k &= B_{PE} J_{2\oplus}\{\hat{\mathbf{x}}_{PE}, 0\} B_E^T \\
G_k &= -B_{PE} J_{1\oplus}\{0, \hat{\mathbf{x}}_{PE}\} B_P^T
\end{aligned}
\tag{B.4}
$$

Assuming that:

$$
\begin{aligned}
\hat{\mathbf{x}}_{PE} &= (\hat{x}_{PE}, \hat{y}_{PE}, \hat{\phi}_{PE})^T \\
S_k &= C_P = \begin{bmatrix} \sigma_y^2 & \sigma_{y\phi} \\ \sigma_{y\phi} & \sigma_\phi^2 \end{bmatrix}
\end{aligned}
$$

According to eq. (B.4) we have:

$$
\begin{aligned}
\mathbf{h}_k &= \hat{y}_{PE} \\
H_k &= \begin{bmatrix} \sin\hat{\phi}_{PE} & \cos\hat{\phi}_{PE} \end{bmatrix} \\
G_k &= \begin{bmatrix} -1 & -\hat{x}_{PE} \end{bmatrix} \\
F_k &= \frac{1}{d} \begin{bmatrix} \sin^2\hat{\phi}_{PE} & \sin\hat{\phi}_{PE}\cos\hat{\phi}_{PE} \\ \sin\hat{\phi}_{PE}\cos\hat{\phi}_{PE} & \cos^2\hat{\phi}_{PE} \end{bmatrix} \\
N_k &= \frac{\hat{y}_{PE}}{d} (\sin\hat{\phi}_{PE},\; \cos\hat{\phi}_{PE})^T
\end{aligned}
$$

where :

$$
d = \sigma_y^2 + 2\,\hat{x}_{PE}\,\sigma_{y\phi} + \hat{x}_{PE}^2\,\sigma_\phi^2
$$

Figure B.2: References involved in the integration of a monocular projection ray the estimation of the robot location.

B.1.2 Estimating the location of a mobile robot from monocular 2D edges

Let $\mathbf{L}_{WR} = (\hat{\mathbf{x}}_{WR}, \hat{\mathbf{p}}_R, C_R, I_3)$ be the estimated location of a mobile robot. Let $\mathbf{x}_{WM}$ represent the location of a corner according to an *a priori* map. Let $\mathbf{L}_{RP} = (\hat{\mathbf{x}}_{RP}, \hat{\mathbf{p}}_P, C_P, B_P)$ be the estimated location of the edge according to a monocular vision. Using only one image, our observation of the point is a projection line between the point and the camera (fig. B.2). Assuming that observation P corresponds to model feature M, we may use it to improve the estimation of the robot location. In this case the state to be estimated is represented by the perturbation vector of R, and the measurement by the perturbation vector of P:

$$\begin{aligned}\mathbf{x} &= \mathbf{d}_R \\ \mathbf{y}_k &= \mathbf{p}_P\end{aligned}$$

Since coincidence between P and M can be expressed using the *inverse constraint* we have:

$$\begin{aligned}\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) &= B_{PM}\mathbf{x}_{PM} \\ &= B_{PM}(\ominus \mathbf{x}_{RP} \ominus \mathbf{x}_{WR} \oplus \mathbf{x}_{WM}) \\ &= B_{PM}(\ominus(\hat{\mathbf{x}}_{RP} \oplus \mathbf{d}_P) \ominus (\hat{\mathbf{x}}_{WR} \oplus \mathbf{d}_R) \oplus \mathbf{x}_{WM}) \\ &= B_{PM}(\ominus \mathbf{d}_P \ominus \hat{\mathbf{x}}_{RP} \ominus \mathbf{d}_R \ominus \hat{\mathbf{x}}_{WR} \oplus \mathbf{x}_{WM}) \\ &= B_{PM}(\ominus \mathbf{d}_P \ominus \hat{\mathbf{x}}_{RP} \ominus \mathbf{d}_R \oplus \hat{\mathbf{x}}_{RM}) \\ &= B_{PM}(\ominus \mathbf{d}_P \ominus \hat{\mathbf{x}}_{RP} \oplus \hat{\mathbf{x}}_{RM} \ominus J_{MR}\mathbf{d}_R) \\ &= B_{PM}(\ominus \mathbf{d}_P \oplus \hat{\mathbf{x}}_{PM} \ominus J_{MR}\mathbf{d}_R) \\ &= B_{PM}(\ominus B_P^T \mathbf{p}_P \oplus \hat{\mathbf{x}}_{PM} \ominus J_{MR}\mathbf{d}_R) \\ &= B_{PM}(\ominus B_P^T \mathbf{y}_k \oplus \hat{\mathbf{x}}_{PM} \ominus J_{MR}\mathbf{x}) \\ &= 0\end{aligned}$$

In this case we have:

$$B_M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \;;\; B_P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \;;\; B_{PM} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

If we consider both estimations centered, we have:

$$\begin{aligned}\mathbf{h}_k &= B_{PM}\hat{\mathbf{x}}_{PM} \\ H_k &= -B_{PM}J_{2\oplus}\{\hat{\mathbf{x}}_{PM}, 0\}J_{MR} \\ G_k &= -B_{PM}J_{1\oplus}\{0, \hat{\mathbf{x}}_{PM}\}B_P^T\end{aligned} \tag{B.5}$$

Assuming that:

Figure B.3: References involved in the integration of a monocular 2D edge to the estimation of the robot location using an imprecise map.

$$\begin{aligned}
\hat{\mathbf{x}}_{PM} &= (\hat{x}_{PM}, \hat{y}_{PM}, \hat{\phi}_{PM})^T \\
\hat{\mathbf{x}}_{MR} &= (\hat{x}_{MR}, \hat{y}_{MR}, \hat{\phi}_{MR})^T \\
S_k &= C_P = \begin{bmatrix} \sigma_y^2 & \sigma_{y\phi} \\ \sigma_{y\phi} & \sigma_\phi^2 \end{bmatrix}
\end{aligned}$$

according to eq. (B.5) we have:

$$\begin{aligned}
\mathbf{h}_k &= \hat{y}_{PM} \\
H_k &= \begin{bmatrix} a & b & c \end{bmatrix} \\
G_k &= \begin{bmatrix} 1 & \hat{x}_{PM} \end{bmatrix} \\
F_k &= \begin{bmatrix} a^2 & a\,b & a\,c \\ a\,b & b^2 & b\,c \\ a\,c & b\,c & c^2 \end{bmatrix} \\
N_k &= \frac{\hat{y}_{PM}}{d}\,(a,\, b,\, c)^T
\end{aligned}$$

where:

$$\begin{aligned}
a &= \sin(\hat{\phi}_{PM} + \hat{\phi}_{MR}) \\
b &= \cos(\hat{\phi}_{PM} + \hat{\phi}_{MR}) \\
c &= \hat{y}_{MR} \sin\hat{\phi}_{PM} - \hat{x}_{MR} \cos\hat{\phi}_{PM} \\
d &= \sigma_y^2 + 2\,\hat{x}_{PM}\,\sigma_{y\phi} + \hat{x}_{PM}^2\,\sigma_\phi^2
\end{aligned}$$

B.1.3 Estimating the location of a mobile robot from monocular 2D edges using an imprecise map

Let $\mathbf{L}_{WR} = (\hat{\mathbf{x}}_{WR}, \hat{\mathbf{p}}_R, C_R, I_3)$ be the estimated location of a mobile robot. Let $\mathbf{L}_{WM} = (\hat{\mathbf{x}}_{WM}, \hat{\mathbf{p}}_M, C_M, B_M)$ represent the location of a corner according to an *imprecise a priori* map. Let $\mathbf{L}_{RP} = (\hat{\mathbf{x}}_{RP}, \hat{\mathbf{p}}_P, C_P, B_P)$ be a monocular 2d edge, a *partial* observation of the corner according to a monocular vision system (fig. B.3). Assuming that observation P corresponds to model feature M, we may use it to improve the estimation of the robot location. In this case the state to be estimated is represented by the perturbation vector of R, and the measurement by the perturbation vector of P:

$$\begin{aligned}
\mathbf{x} &= \mathbf{d}_R \\
\mathbf{y}_k &= \mathbf{p}_P
\end{aligned}$$

Again, coincidence between P and M can be expressed using the *inverse constraint*. Thus we have:

$$\begin{aligned}
\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) &= B_{PM}\mathbf{x}_{PM} \\
&= B_{PM}(\ominus\mathbf{x}_{RP} \ominus \mathbf{x}_{WR} \oplus \mathbf{x}_{WM}) \\
&= B_{PM}(\ominus(\hat{\mathbf{x}}_{RP} \oplus \mathbf{d}_P) \ominus (\hat{\mathbf{x}}_{WR} \oplus \mathbf{d}_R) \oplus \hat{\mathbf{x}}_{WM} \oplus \mathbf{d}_M) \\
&= B_{PM}(\ominus\mathbf{d}_P \ominus \hat{\mathbf{x}}_{RP} \ominus \mathbf{d}_R \ominus \hat{\mathbf{x}}_{WR} \oplus \hat{\mathbf{x}}_{WM} \oplus \mathbf{d}_M) \\
&= B_{PM}(\ominus\mathbf{d}_P \ominus \hat{\mathbf{x}}_{RP} \ominus \mathbf{d}_R \oplus \hat{\mathbf{x}}_{RM} \oplus \mathbf{d}_M) \\
&= B_{PM}(\ominus\mathbf{d}_P \ominus \hat{\mathbf{x}}_{RP} \oplus \hat{\mathbf{x}}_{RM} \ominus J_{MR}\mathbf{d}_R \oplus \mathbf{d}_M) \\
&= B_{PM}(\ominus\mathbf{d}_P \oplus \hat{\mathbf{x}}_{PM} \ominus J_{MR}\mathbf{d}_R \oplus \mathbf{d}_M) \\
&= B_{PM}(\ominus B_P^T\mathbf{p}_P \oplus \hat{\mathbf{x}}_{PM} \ominus J_{MR}\mathbf{d}_R \oplus B_M^T\mathbf{p}_M) \\
&= 0
\end{aligned}$$

If we consider both estimations centered, we have:

$$\begin{aligned}
\mathbf{h}_k &= B_{PM}\hat{\mathbf{x}}_{PM} \\
H_k &= -B_{PM}J_{2\oplus}\{\hat{\mathbf{x}}_{PM}, 0\}J_{MR} \\
G_k &= -B_{PM}J_{1\oplus}\{0, \hat{\mathbf{x}}_{PM}\}B_P^T
\end{aligned} \tag{B.6}$$

In this case we have:

$$B_M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \;;\; B_P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \;;\; B_{PM} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

Assuming that edge P corresponds to model point M, we use it to improve the estimation of the robot location by means of the *inverse constraint*. If we consider both estimations centered, and assuming that:

$$\begin{aligned}
\hat{\mathbf{x}}_{PM} &= (\hat{x}_{PM}, \hat{y}_{PM}, \hat{\phi}_{PM})^T \\
\hat{\mathbf{x}}_{MR} &= (\hat{x}_{MR}, \hat{y}_{MR}, \hat{\phi}_{MR})^T \\
S_k &= C_P = \begin{bmatrix} \sigma_y^2 & \sigma_{y\phi} \\ \sigma_{y\phi} & \sigma_\phi^2 \end{bmatrix}
\end{aligned}$$

according to eq. (B.6) we have:

$$\begin{aligned}
\mathbf{h}_k &= \hat{y}_{PM} \\
H_k &= \begin{bmatrix} a & b & c \end{bmatrix} \\
G_k &= \begin{bmatrix} 1 & \hat{x}_{PM} \end{bmatrix} \\
F_k &= \begin{bmatrix} a^2 & a\,b & a\,c \\ a\,b & b^2 & b\,c \\ a\,c & b\,c & c^2 \end{bmatrix} \\
N_k &= \frac{\hat{y}_{PM}}{d}\,(a,\, b,\, c)^T
\end{aligned}$$

where:

$$\begin{aligned} a &= \sin(\hat{\phi}_{PM} + \hat{\phi}_{MR}) \\ b &= \cos(\hat{\phi}_{PM} + \hat{\phi}_{MR}) \\ c &= \hat{y}_{MR} \sin\hat{\phi}_{PM} - \hat{x}_{MR} \cos\hat{\phi}_{PM} \\ d &= \sigma_y^2 + 2\,\hat{x}_{PM}\,\sigma_{y\phi} + \hat{x}_{PM}^2\,\sigma_\phi^2 \end{aligned}$$

B.2 The Extended Kalman Filter

In some cases, it is more convenient to use the extended Kalman Filter than the extended Information Filter. The extended Kalman Filter can be stated as follows: again let $\mathbf{x}$ be the state vector whose value is to be estimated, and let there be n independent and possibly partial observations $\mathbf{y}_k$ of $\mathbf{x}$, where $k \in \{1, \ldots, n\}$, affected by white Gaussian noise:

$$\hat{\mathbf{y}}_k = \mathbf{y}_k + \mathbf{u}_k \;\;;\;\; \mathbf{u}_k \sim N(0, S_k)$$

Let each observation $\mathbf{y}_k$ be related to $\mathbf{x}$ by an implicit nonlinear function of the form $\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) = 0$. We use a first order approximation of $\mathbf{f}_k$:

$$\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) \simeq \mathbf{h}_k + H_k(\mathbf{x} - \hat{\mathbf{x}}) + G_k(\mathbf{y}_k - \hat{\mathbf{y}}_k)$$

where:

$$\mathbf{h}_k = \mathbf{f}_k(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k) \;\;;\;\; H_k = \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k)} \;\;;\;\; G_k = \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{y}}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k)} \tag{B.7}$$

Let $\hat{\mathbf{x}}_{k-1}$ and P_{k-1} be the estimate of the state vector and its covariance after integrating the first k-1 observations. The estimate $\hat{\mathbf{x}}_k$ of the state vector and its covariance P_k after integrating measurement $\hat{\mathbf{y}}_k$ are:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} - K_k H_k \hat{\mathbf{x}}_{k-1} \;\;;\;\; P_k = P_{k-1} - K_k H_k P_{k-1} \tag{B.8}$$

where:

$$K_k = P_{k-1} H_k (H_k P_{k-1} H_k^T + G_k S_k G_k^T)^{-1} \tag{B.9}$$

The case of interest for the application of this filter in this work is given below.

B.2.1 Estimating the location of a mobile robot and map corners from monocular 2D edges

Let $\mathbf{L}_{WR} = (\hat{\mathbf{x}}_{WR}, \hat{\mathbf{p}}_R, C_R, I_3)$ be the estimated location of a mobile robot. Let $\mathbf{L}_{WM_i} = (\hat{\mathbf{x}}_{WM_i}, \hat{\mathbf{p}}_{M_i}, C_{M_i}, B_{M_i})$ represent the estimated location of an corner, where $i \in \{1, \ldots, m\}$. Let $\mathbf{L}_{RP} = (\hat{\mathbf{x}}_{RP}, \hat{\mathbf{p}}_P, C_P, B_P)$ be a monocular 2D edge, a *partial* observation of corner $M_k = M$ according to a monocular vision system (fig. B.4). Assuming that observation P corresponds to model feature M, we may use it to improve the estimation of the robot location, as well as the location of the feature itself. In this case, the state to be estimated

Figure B.4: References involved in the integration of a subfeature the estimation of the robot location.

is represented by the perturbation vectors of R and M, and the measurement by the perturbation vector of P:

$$
\begin{aligned}
\mathbf{x} &= [\mathbf{d}_R \, \mathbf{p}_M]^T \\
\mathbf{y}_k &= \mathbf{p}_P
\end{aligned}
$$

Since coincidence between P and M can be expressed using the *inverse constraint* we have:

$$
\begin{aligned}
\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) &= B_{PM}\mathbf{x}_{PM} \\
&= B_{PM}(\ominus \mathbf{x}_{RP} \ominus \mathbf{x}_{WR} \oplus \mathbf{x}_{WM}) \\
&= B_{PM}(\ominus(\hat{\mathbf{x}}_{RP} \oplus \mathbf{d}_P) \ominus (\hat{\mathbf{x}}_{WR} \oplus \mathbf{d}_R) \oplus \hat{\mathbf{x}}_{WM} \oplus \mathbf{d}_M) \\
&= B_{PM}(\ominus \mathbf{d}_P \ominus \hat{\mathbf{x}}_{RP} \ominus \mathbf{d}_R \ominus \hat{\mathbf{x}}_{WR} \oplus \hat{\mathbf{x}}_{WM} \oplus \mathbf{d}_M) \\
&= B_{PM}(\ominus \mathbf{d}_P \ominus \hat{\mathbf{x}}_{RP} \ominus \mathbf{d}_R \oplus \hat{\mathbf{x}}_{RM} \oplus \mathbf{d}_M) \\
&= B_{PM}(\ominus \mathbf{d}_P \ominus \hat{\mathbf{x}}_{RP} \oplus \hat{\mathbf{x}}_{RM} \ominus J_{MR}\mathbf{d}_R \oplus \mathbf{d}_M) \\
&= B_{PM}(\ominus \mathbf{d}_P \oplus \hat{\mathbf{x}}_{PM} \ominus J_{MR}\mathbf{d}_R \oplus \mathbf{d}_M) \\
&= B_{PM}(\ominus B_P^T \mathbf{p}_P \oplus \hat{\mathbf{x}}_{PM} \ominus J_{MR}\mathbf{d}_R \oplus B_M^T \mathbf{p}_M) \\
&= 0
\end{aligned}
$$

If we consider both estimations centered, we have:

$$
\begin{aligned}
\mathbf{h}_k &= B_{PM}\hat{\mathbf{x}}_{PM} \\
H_R &= \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{d}_R}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k)} = -B_{PM}J_{2\oplus}\{\hat{\mathbf{x}}_{PM}, 0\}J_{MR} \\
H_M &= \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{p}_M}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k)} = B_{PM}J_{2\oplus}\{\hat{\mathbf{x}}_{PM}, 0\}B_M^T \\
G_k &= -B_{PM}J_{1\oplus}\{0, \hat{\mathbf{x}}_{PM}\}B_P^T
\end{aligned}
\tag{B.10}
$$

In this case we have:

$$
B_M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \; ; \; B_P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \; ; \; B_{PM} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}
$$

If we consider both estimations centered, and assuming that:

$$
\begin{aligned}
\hat{\mathbf{x}}_{PM} &= (\hat{x}_{PM}, \hat{y}_{PM}, \hat{\phi}_{PM})^T \\
\hat{\mathbf{x}}_{MR} &= (\hat{x}_{MR}, \hat{y}_{MR}, \hat{\phi}_{MR})^T \\
S_k &= C_P = \begin{bmatrix} \sigma_y^2 & \sigma_{y\phi} \\ \sigma_{y\phi} & \sigma_\phi^2 \end{bmatrix}
\end{aligned}
$$

according to eq. (B.10) we have:

$$
\begin{aligned}
\mathbf{h}_k &= \hat{y}_{PM} \\
H_R &= \begin{bmatrix} a & b & c \end{bmatrix} \\
H_M &= \begin{bmatrix} \sin\hat{\phi}_{PE} & \cos\hat{\phi}_{PE} \end{bmatrix} \\
H_k &= [H_R\, 0 \ldots 0\; H_M\, 0 \ldots 0]^T \\
G_k &= \begin{bmatrix} 1 & \hat{x}_{PM} \end{bmatrix}
\end{aligned}
$$

where:

$$
\begin{aligned}
a &= \sin(\hat{\phi}_{PM} + \hat{\phi}_{MR}) \\
b &= \cos(\hat{\phi}_{PM} + \hat{\phi}_{MR}) \\
c &= \hat{y}_{MR} \sin\hat{\phi}_{PM} - \hat{x}_{MR} \cos\hat{\phi}_{PM}
\end{aligned}
$$

Note that in this case the state to be estimated is composed of the perturbation vectors of the robot and *all* the map features:

$$
\begin{aligned}
\mathbf{x} &= [\mathbf{d}_R\, \mathbf{p}_1 \ldots \mathbf{p}_m]^T \\
P_{\mathbf{x}} &= [P_{i,j}]
\end{aligned}
$$

Where $P_{0,0} = C_R$, $P_{i,i} = C_{M_i}$ and $P_{i,j}$ expresses the correlation between elements i and j of $\mathbf{x}$. This is scheme is usually referred to as *stochastic map*.

Appendix C

Tracking

In this section we describe the mathematical details of the tracking mechanism used to solve the correspondence problem. Tracking is used to estimate the motion of vertical edges in the image sequence (e.g. the variation of location parameters and edge length). This estimation is used to predict where each edge should be located in the next image, and thus solve the correspondence problem by choosing the edge nearest to the predicted location in each case.

C.1 Kalman Filter

The Kalman filter allows to estimate the state of a linear dynamic system using a set of measurements of this state. Both the system dynamics and the measurements are subject to noise and imprecision. In this case we will use the Gauss-Markov model for systems discretized in time, which is defined by the following difference equations:

$$\begin{array}{lcll} \mathbf{x}_{k+1} & = & F_k \mathbf{x}_k + \mathbf{w}_k & \text{dynamic model} \\ \mathbf{z}_k & = & H_k \mathbf{x}_k + \mathbf{v}_k & \text{measurement model} \end{array}$$

where $\mathbf{x}_k$ is the *system state vector* at instant t_k, $\mathbf{z}_k$ is the *measurement*, $\mathbf{w}_k$ the *input noise* and $\mathbf{v}_k$ the *measurement noise*. Both sources of noise are assumed to be white, Gaussian, and independent, with zero mean and known covariance (Q_k and R_k, respectively):

$$\begin{array}{ll} E[\mathbf{v}_k] = 0 & E[\mathbf{v}_k \ \mathbf{v}_l^T] = \delta_{kl} \ R_k \\ E[\mathbf{w}_k] = 0 & E[\mathbf{w}_k \ \mathbf{w}_l^T] = \delta_{kl} \ Q_k \\ E[\mathbf{v}_k \ \mathbf{w}_l^T] = 0 & \end{array}$$

Term δ_{kl} is deniminated the delta of Kronecker, whose value is 1 when $k = l$, and 0 otherwise. The initial state of the system $\mathbf{x}_0$ is considered a Gaussian random variable, independent of both $\mathbf{v}_k$ y $\mathbf{w}_k$ (for every instant k), with mean $\hat{\mathbf{x}}_0$ and known covariance P_0.

Given a set of measurements $Z_k = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$, the Kalman filter allows to *recursively* obtain a linear estimation of minimum variance of the system state in an instant k, as well as the covariance of the estimation error. Each estimation cycle (for every instant k) is composed of two phases:

1. *Prediction*
   Using the *dynamic model of the system*, and given the estimation of the state at instant $k-1$ ($\hat{\mathbf{x}}_{k|k-1}$), a first estimation of the state at instant k is calculated

2. *Estimation*
   A new estimation ($\hat{\mathbf{x}}_{k|k}$) is calculated using the measurement obtained at instant k, by means of the *measurement model*

If we can assume that the time elapsed between instant k and $k+1$ is the same, we can adopt the following notation:

$$\begin{array}{l}
\hat{\mathbf{x}}_p = \hat{\mathbf{x}}_{k|k-1} = E[\mathbf{x}_k | Z_{k-1}] \\
P_p = E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T] = Cov(\tilde{\mathbf{x}}_{k|k-1}) \\
\text{con } \tilde{\mathbf{x}}_{k|k-1} = \mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}
\end{array}$$

$$\begin{array}{l}
\hat{\mathbf{x}}_e = \hat{\mathbf{x}}_{k|k} = E[\mathbf{x}_k | Z_k] \\
P_e = E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T] = Cov(\tilde{\mathbf{x}}_{k|k}) \\
\text{con } \tilde{\mathbf{x}}_{k|k} = \mathbf{x}_k - \hat{\mathbf{x}}_{k|k}
\end{array}$$

In this case the Kalman filter can be stated as follows: given the *estimation* of the system state ($\hat{\mathbf{x}}_e$, P_e) at some instant, the *prediction* ($\hat{\mathbf{x}}_p$, P_p) at the next instant is calculated as follows:

$$\begin{array}{rcl}
\hat{\mathbf{x}}_p & = & F\,\hat{\mathbf{x}}_e \\
P_p & = & F\,P_e F^T + Q
\end{array} \tag{C.1}$$

Given a new measurement $(\mathbf{z}, R)$, the new *estimation* of the system is calculated as follows:

$$\begin{array}{rcl}
S & = & HP_pH^T + R \\
K & = & P_pH^TS^{-1} \\
\mathbf{r} & = & \mathbf{z} - H\hat{\mathbf{x}}_p \\
\hat{\mathbf{x}}_e & = & \hat{\mathbf{x}}_p + K\mathbf{r} \\
P_e & = & (I - KH)P_p
\end{array} \tag{C.2}$$

Vector $\mathbf{r}$ is denominated *prediction measurement error* or *innovation*; S is its covariance. To determine whether measurement $\mathbf{z}$ can be considered compatible with the prediction, a statistical measurement, the *Mahalanobis* distance, is used:

$$d = \mathbf{r}^T S^{-1} \mathbf{r}$$

In the Gaussian case, the Mahalanobis distance follows a χ^2_n distribution, where n denotes the degrees of freedom (n is the number of parameters measured). Given a measurement $(\mathbf{z}, R)$, whose innovation for a given estimation is $(\mathbf{r}, S)$, and given an acceptance level α, the measurement can be considered compatible with the state if:

$$\left\{ d \leq \chi^2_n(\alpha) \right\}$$

This means that for example if $\alpha = 0.05$, a measurement satisfying the previous test will be spurious only in 5% of the cases.

C.2 Application to edge tracking

We consider three parameters in the identification of each edge:

- x, the X coordinate of the medium point
- y, the Y of the medium point
- l, the edge length

Edge orientation is not used since only vertical edges are considered. Given that we intend to determine the variation of these parameters in the image sequence, in this case the state vector of the system is:

$$\mathbf{x} = \left(x,\, \dot{x},\, y,\, \dot{y},\, l,\, \dot{l}\right)^T$$

Assuming that each parameter p is decoupled from the rest, and assuming also that its variation is constant, the state equation for each pair $(p, \dot{p})^T$ is defined in the following way:

$$\begin{aligned} p_p &= p_e + \dot{p}\Delta t \\ \dot{p}_p &= \dot{p}_e \end{aligned}$$

where Δt is the elapsed time between the acquisition of an image and the acquisition of the next. From the prediction equation (C.1), we conclude that:

$$F_p = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

and given that we consider the parameters uncoupled:

$$F = \begin{bmatrix} F_x & 0 & 0 & 0 \\ 0 & F_y & 0 & 0 \\ 0 & 0 & F_\phi & 0 \\ 0 & 0 & 0 & F_l \end{bmatrix}$$

Given that in each image we can observe each parameter p, but not its derivate $\dot{p}$, from equation (C.2) we have that the state and the measurement are related by:

$$H_p = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Again, considering the parameters uncoupled:

$$H = \begin{bmatrix} H_x & 0 & 0 & 0 \\ 0 & H_y & 0 & 0 \\ 0 & 0 & H_\phi & 0 \\ 0 & 0 & 0 & H_l \end{bmatrix}$$

To model the covariance of the state noise Q, we use the following discretization model:

$$Q_p = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \Delta t \end{bmatrix} \dot{q}_p$$

and thus we will have:

$$Q = \begin{bmatrix} Q_x & 0 & 0 & 0 \\ 0 & Q_y & 0 & 0 \\ 0 & 0 & Q_\phi & 0 \\ 0 & 0 & 0 & Q_l \end{bmatrix}$$

When the images are obtained at equal intervals of time, we may have $\Delta t = 1$.

Bibliography

[1] C.A. Bozzo. *Le Filtrage Optimal et ses Applications aux Problèmes de Poursuite.* Technique et Documentation, Montrouge, France, 1983.

[2] J. Castellanos. The importance of maintaining correlations in the representation of the environment of a mobile robot. Research report, LSR Technishe Universitaet Muenchen, Munich, Germany, 1996.

[3] M. Devy and H. Bulata. Landmark-based vs. feature-based localization of a mobile robot in a structured environment. In *Proc. 7th Int. Conf. on Advanced Robotics, ICAR 95*, pages 998–1007, Catalonia, Spain, September 1995.

[4] A. Gelb, editor. *Applied Optimal Estimation.* The M.I.T. Press, Cambridge, Mass., 1974.

[5] P. Herbért, S. Betgé-Brezetz, and R. Chatila. Probabilistic map learning: Necessity and difficulties. In Leo Dorst, Michiel van Lambalgen, and Frans Voorbraak, editors, *Reasoning with Uncertainty in Robotics*, pages 307–321. Springer, 1996.

[6] J. Horn and G. Schmidt. Continuous Localization of a Mobile Robot Based on 3D–Laser–Range–Data, Predicted Sensor Images, and Dead–Reckoning. *Robotics and Autonomous Systems, Special Issue "Research on Autonomous Mobile Robots"*, 14:99–118, 1995.

[7] X. Lebègue and J.K. Aggarwal. Significant line segments for an indoor mobile robot. *IEEE Trans. on Robotics and Automation*, 9(6):801–815, 1993.

[8] J.J. Leonard and H.F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation.* Kluwer Academic Publishers, London, 1992.

[9] J.J. Leonard, H.F. Durrant-Whyte, and I.J. Cox. Dynamic map building for an autonomous mobile robot. *IEEE Journal of Robotics Research*, 11(4), 1992.

[10] J. Neira. *Geometric Object Recognition in Multisensor Systems.* PhD thesis, Dpto. de Ingeniería Eléctrica e Informática, University of Zaragoza, Spain, September 1993.

[11] J. Neira, J. Horn, J.D. Tardós, and G. Schmidt. Multisensor mobile robot localization. In *IEEE International Conference on Robotics and Automation*, Minneapolis, USA, May 1996.

[12] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In J.F. Lemmer and L.N. Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 435–461. Elsevier Science Pub., 1988.

[13] J.D. Tardós. *Integración Multisensorial para Reconocimiento y Localización de Objetos en Robótica.* PhD thesis, Dpto. de Ingeniería Eléctrica e Informática, University of Zaragoza, Spain, Febrero 1991.

[14] Z. Zhang and O.D. Faugeras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *International Journal of Computer Vision*, 3(7):211–241, 1992.