Geometric Object Recognition in Multisensor Systems

José Neira Parra

Tesis Doctoral

Departamento de Ingeniería Eléctrica e Informática
Centro Politécnico Superior de Ingenieros
Universidad de Zaragoza

Agosto 1993

Contents

Chapter 1

Multisensor Object Recognition

1.1 Introduction

Object recognition is one of the fundamental areas of research in intelligent robotics. Identifying and locating objects on the workspace of a robot is considered a fundamental problem to solve in order to develop more flexible and powerful robotic systems. In industrial robotic tasks, the perirobotic engineering required to structure the robot environment has become increasingly cost-ineffective. Furthermore, robots are being introduced in unstructured and less predictable environments, such as indoor and outdoor exploration. Thus, it is necessary to improve their ability to perceive and interpret their workspace.

Actually, there is a great variety of sensing devices that can be used to extract environment information, such as vision systems, infrared, ultrasound and laser proximity sensors, force and torque sensors, structured light systems and many more. Nevertheless, sensors give *partial*, *inaccurate* and *error-prone* information. The use of several sensors allows to overcome these limitations, as individual sensing capabilities are sinergistically complemented and redundancy in multisensorial information allows the system to be more robust to sensor error or malfunction.

Sensors can obtain parametric and geometric information. Parametric descriptors are *global* properties of objects, such as their color, texture, and form descriptors. *Parametric recognition* is generally stated as a statistical classification problem, and can be very efficient in some situations, where objects can be found isolated and in predetermined poses. Unfortunately, it is very unrobust in situations where objects are partially occluded and jumbled together. These types of scenes can be analyzed more adequately using *geometric recognition.* Geometric information is related to *local features* of the object surface, such as edges, planar surfaces, pegs, holes, etc. Geometric recognition is based on the search for correspondences between a set of sensed features and a database of geometric models of the objects. There are two factors that make geometric recognition a difficult problem:

- Information extracted from the environment is generally low level, subject to *uncertainty in measurement* and *ambiguity in interpretation.*
- Unstructured environments confronts us with an *exponential solution space*, even if only a limited set of known objects (that is, where detailed models are available) is considered.

Uncertainty is introduced by *sensors* that give inaccurate, and sometimes spurious measurements; by *data processing algorithms and methods* that can be approximate and thus introduce errors; and by *the observation-model matching process*, whose performance is affected by occlusion and similarities between object models, or between different poses of objects. Uncertainty can be classified into two categories:

- *Quantitative uncertainty.* It is related to the imprecision of geometric information such as object or feature location, and to measured properties of geometric entities such as dimensions, distances, and angles. This kind of uncertainty is normally introduced by inaccurate sensor measurements during the data acquisition process, and by the use of approximate methods.
- *Qualitative uncertainty.* It refers to the validity of an interpretation of sensor observations. Since recognition is a matching process, an observation can have many potential pairings that would support different hypotheses on the object identity and location. (An observation can also be spurious.) Therefore, qualitative uncertainty is related to the symbolic association of observations to models in the recognition process.

Reduction in the exponential complexity of geometric recognition can be achieved by the application of two fundamental ideas: the use of *validation mechanisms* that allow the system to discard entire subspaces of the solution space from further consideration, and the use of *strategies for the generation* *and verification of hypotheses*, that can help the system in searching the solution space more efficiently to obtain more plausible hypotheses promptly.

The fundamental contributions of this work are the definition of such validation mechanisms and hypothesis management strategies, making an explicit consideration of uncertainty. The fundamental objective of our work is *generality*: we are concerned with the development recognition mechanisms that can be applied to any combination of sensors or geometric features, making them adequate for their use in multisensor systems. In the next sections we formally define the recognition problem, and have a closer look at the currently proposed recognition schemes, in order to precisely identify which aspects of recognition constitute the contribution of the present work.

1.2 Problem Definition

Formally, the multisensor recognition problem can be stated in the following way: let $\mathcal{E} = \{e_1, \ldots, e_s\}$ be a set of s diverse sensorial observations of geometric features of the surface of the objects present in the scene, obtained by different sensors. Let $\mathcal{O} = \{O_1, \ldots, O_r\}$ be a set of r object models that the system is capable of recognizing. Let $\mathcal{M} = \{m_1, \ldots, m_n\}$ be the set of model features of all object models. Each object model O_i has an associated set of geometric features, $O_i = \{m_{i_1}, \ldots, m_{i_{n_i}}\} \subseteq \mathcal{M}$. (For simplicity in notation, the model and its set of features are denoted with the same letter.)

The goal of the recognition system is to generate an interpretation, in terms of the identity and localization of objects, that best fits the set of observations. Such an interpretation relates each observation e_j with a model feature m_k by means of a *pairing* $p_i = (e_j, m_k)$. Let $\mathcal{P}$ denote the set of all pairings. An interpretation is a set of object-location hypotheses $\mathcal{H} = \{h_1, \ldots, h_h\}$, where each hypothesis has the form:

$$h_i = \{\text{Instance of } O_j \text{ at } \mathbf{L}_{O_{h_i}},\ \mathcal{S}_{h_i},\ \mathcal{C}_{h_i},\ \mathcal{U}_{h_i}\}$$

where $\mathbf{L}_{O_{h_i}}$ is the hypothesized object location, the set $\mathcal{S}_{h_i} \subseteq \mathcal{P}$ is the set of pairings which support the hypothesis, $\mathcal{C}_{h_i} \subseteq O_j$ is the set of features whose location in the scene has been predicted but were not found, and constitute the hypothesis counterevidence, and $\mathcal{U}_{h_i} \subseteq O_j$ is the set of model features that have not been verified, and constitute its potential support.

The space of solutions is usually represented by means of an *interpretation tree* [Grimson 87] (fig. 1.1). This tree has s levels; each node at a level k, called a *k-interpretation*, provides an interpretation for the first k sensorial observations. Each node of the tree has $n + 1$ branches, corresponding to

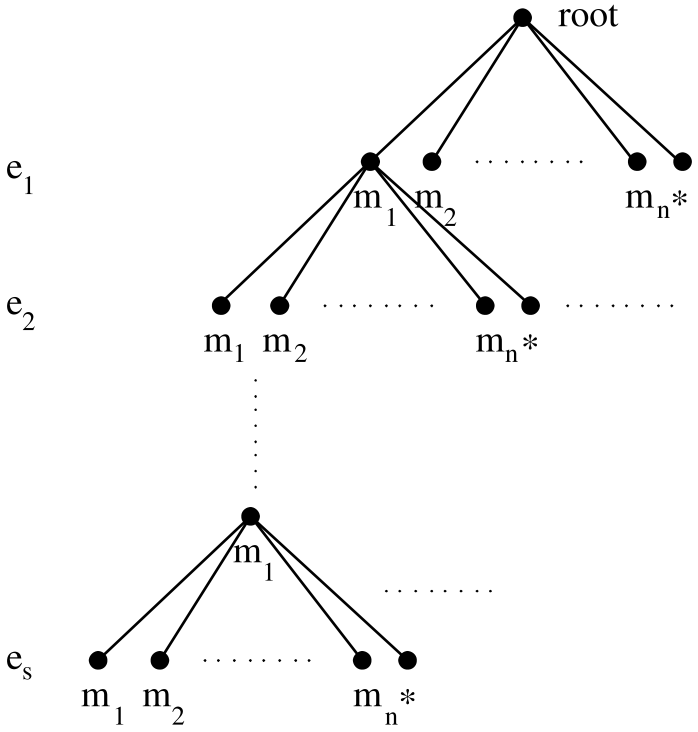


Figure 1.1: Interpretation tree that represents the solution space

each of the alternative interpretations for the observation k (including the possibility that the observation be spurious, denoted by *). Given s observations and n model features, and considering that a sensorial observation can be spurious, we have that there are $(n+1)^s$ terminal nodes, or leaves, in the interpretation tree. That is, the number of possible interpretations for the full set of observations grows exponentially with s.

The recognition of an object implies carrying out two tasks: determining the *identity* of the object, and computing its *location* in the scene. Object identification is a search problem, while computing its location is an estimation problem. This twofold goal has given rise to two fundamental recognition schemes:

- The *identifying before locating* scheme [Grimson 90a], where the identity of the object is determined first, and then its location in the scene is estimated. In this scheme, the validation of a hypothesis is done using *location independent constraints*, a set of simple and fast validations of geometric relations between features, that are independent of object location.

- The *identifying while locating* scheme [Faugeras 86], in which identification and localization are carried out simultaneously. The availability of an estimated location of the object during the identification process allows to use *location dependent constraints*, which constitute a powerful validation mechanism.

There are some basic aspects of the recognition problem that are common to both schemes, and some others that are substantially different. In the following sections we will describe these schemes in a top-down fashion, in order to identify their similarities and differences.

1.3 The Identifying before Locating Scheme

The *identifying before locating* scheme is based on separating the processes of determining the identity of the object, and of determining its location in the scene. In algorithm 1.1, we give a basic and simple implementation of this approach, where one object model is considered[1]. Function `search_for_pairings` returns a set of hypotheses, where the hypothesized location of the object has *not* been estimated. Function `locate_and_validate` performs this estimation with the purpose of determining which hypotheses are consistent. Let us study each of these steps in detail.

1.3.1 Searching for pairings

Given that identification is a search problem of exponential complexity, the idea behind the *identifying before locating* scheme is to use very simple and fast validation mechanisms to determine whether a given hypothesis is consistent with the set of observations. Such validations can be made using *geometric constraints*. Geometric constraints are a set of parameters that derive from the dimensions of the features and from their relative location. If they involve only one feature, they are denominated *unary constraints*. For example, an observed edge of length l_e can only correspond to a model edge of length $l_m \geq l_e$. It is also possible to validate geometric constraints

[1] This is sufficient for illustrative purposes, and its extension to the consideration of several objects is straightforward.

```
FUNCTION identify_before_locating ($\mathcal{E}$, $\mathcal{M}$)

; $\mathcal{E}$: set of sensorial observations
; $\mathcal{M}$: set of model features

; returns $\mathcal{H}_v$, a set of object-location hypotheses, that are
; consistent interpretations of the set of observations

    $\mathcal{H}_g$ := search_for_pairings ($\emptyset$, $\mathcal{E}$, $\mathcal{M}$);
    $\mathcal{H}_v$ := locate_and_validate ($\mathcal{H}_g$);

RETURN $\mathcal{H}_v$;
END;
```

Algorithm 1.1: *Identifying before Locating*

```
FUNCTION search_for_pairings ($\mathcal{S}_h$, $\mathcal{E}$, $\mathcal{M}$)

; $\mathcal{S}_h$: current set of pairings
; $\mathcal{E}$: remaining observations to be paired
; $\mathcal{M}$: set of candidate model features

; This function selects an observation from $\mathcal{E}$, and uses binary constraints
; to determine which model features of $\mathcal{M}$ can be paired with it.
; For each acceptable pairing, it recurs to accumulate new pairings for $\mathcal{S}_h$

    $\mathcal{H} := \emptyset$;
    IF $\mathcal{E} = \emptyset$ THEN
        $\mathcal{H} := \mathcal{H} \cup \{\mathcal{S}_h\}$;
    ELSE
        $e$ := select_observation ($\mathcal{E}$);                                  (Chapter 6)
        FOR $m \in \mathcal{M}$ DO
            $p := (e, m)$;
            IF satisfy_unary_constraints ($p$) THEN
                $binary$ := TRUE;
                FOR $p_p = (e_p, m_p) \in \mathcal{S}_h$ WHILE $binary$ DO
                    $binary$ := satisfy_binary_constraints ($p_p$, $p$);         (Chapter 3)
                OD;
                IF $binary$ THEN
                    $\mathcal{H} := \mathcal{H} \cup$ search_for_pairings ($\mathcal{S}_h \cup \{p\}$ , $\mathcal{E} \setminus \{e\}$ , $\mathcal{M}$);
                FI;
            FI;
        OD;
        $\mathcal{H} := \mathcal{H} \cup$ search_for_pairings ($\mathcal{S}_h$, $\mathcal{E} \setminus \{e\}$ , $\mathcal{M}$);
    FI;

RETURN $\mathcal{H}$;
END;
```

Algorithm 1.2: *Searching for pairings*

between pairs of features. Such constraints are denominated *binary constraints*. For example, two observed edges may only correspond to two model edges whose relative distance and angle is the same (taking into account sensor precision). These two types of geometric constraints can be computed and validated without having an estimation of the object location. For this reason we call them *location independent constraints*.

The hypothesis generation process is based on traversing the interpretation tree in search for consistent interpretations. In algorithm 1.2[2], this process is written as a recursive procedure in which, at each step of the recursion, all consistent pairings between an observation e and the model features in $\mathcal{M}$ are obtained.

Note that for candidate pairing p, it is verified whether it satisfies the binary constraints with *each* of the pairings already included in the current set of pairings $\mathcal{S}_h$. Thus, the number of binary constraints to verify for a given hypothesis grows *polynomially* with the number of paired observations. Even if computing and validating these geometric constraints is fast (in terms of runtime), this may lead to a great amount of computation.

An important issue regarding this hypothesis generation process is the selection of the observation e that should be processed next (implemented in the `select_observation` function). The most suitable observation is the one that generates as few pairings as possible. This allows to discard incoherent hypotheses in the upper levels of the interpretation tree, eliminating large subtrees from further consideration. For this purpose, the generally accepted criteria is to select *salient* features, features that are less frequent in the models because of their geometric characteristics or their size.

1.3.2 Locating and Validating

The validation of location independent constraints assures only *local* consistency. This means that, in a hypothesis, the geometry of any pair of observations is consistent with their corresponding model features, but this does not assure that the interpretation is *globally* consistent [Grimson 90a]. Therefore, it is necessary to estimate the object location in order to determine whether the location of each observation and that of its corresponding model feature coincide, taking into account sensor precision. This validation is related to the fact that we consider *rigid* objects, that is, we only consider transformations that preserve the distance between points of the object.

[2] This algorithm is equivalent to the one proposed in [Grimson 90a], where it is written in a nonrecursive way.

```
FUNCTION locate_and_validate ($\mathcal{H}$)

; $\mathcal{H}$: set of hypotheses $h$ whose object location has not been estimated
; and contain only the support pairings $\mathcal{S}_h$

; For each hypothesis, the object location is estimated and the global
; consistency is validated using the rigidity constraint

  $\mathcal{H}_v := \emptyset$;
  FOR $\mathcal{S}_h \in \mathcal{H}$ DO
    $\mathbf{L}_h$ := estimate__object_location ($\mathcal{S}_h$);                    (Chapter 2)
    valid := TRUE;
    FOR $p \in \mathcal{S}_h$ WHILE valid DO
      valid := satisfy_rigidity_constraint ($\mathbf{L}_h$, $p$);              (Chapter 3)
    OD;
    IF valid THEN
      $\mathcal{H}_v := \mathcal{H}_v \cup h$;
    FI;
  OD;

RETURN $\mathcal{H}_v$;
END;
```

Algorithm 1.3: *Locating and Validating*

Consequently, it is called the *rigidity* constraint. It is one of several constraints that can be validated only when an estimation of the object location is available, the *location dependent* constraints. Algorithm 1.3 implements this hypotheses validation mechanism.

Object localization is usually carried out using some estimation method that finds a transformation such that the error between each transformed model feature and its corresponding observed feature is minimal in some sense. In [Tardós 92a, Tardós 92b], Tardós develops a general integration mechanism for uncertain geometric information, based on the Extended Kalman and information filters, which we adopt in this work.

1.4 The Identifying while Locating Scheme

The fundamental idea behind the *identifying while locating* recognition scheme is that an estimation of the location of the hypothesized object is a very important source of information for the identification process. For this reason, the complexity of the recognition process can be reduced if identification and localization are performed simultaneously.

```
FUNCTION identify_while_locating ($\mathcal{E}$, $\mathcal{M}$)

; $\mathcal{E}$: set of available observations
; $\mathcal{M}$: set of model features

; Generates hypotheses using two independent observations that allow to
; determine the object location, so that they can be verified using
; location dependent constraints

    REPEAT
        $e_f$ := select_first_observation ($\mathcal{E}$);                                   (Chapter 6)
        $e_s$ := select_second_observation $\left(\mathcal{E} \setminus \left\{e_f\right\}, e_f\right)$;               (Chapter 6)
        $\mathcal{H}_g$ := search_for_pairings $\left(\emptyset, \left\{e_f, e_s\right\}, \mathcal{M}\right)$;
        $\mathcal{H}_v := \emptyset$;
        FOR $h \in \mathcal{H}_g$ DO
            $\mathbf{L}_h$ := estimate_object_location ($\mathcal{S}_h$);                      (Chapter 2)
            $\mathcal{H}_v := \mathcal{H}_v \cup$ verify_hypothesis ($h$, $\mathcal{E}$, $\mathcal{M}$);
            $\mathcal{H}_g := \mathcal{H}_g \setminus \{h\}$;
        OD;
        $\mathcal{E} := \mathcal{E} \setminus \{e_f, e_s\}$;
    UNTIL $\mathcal{H}_v \neq \emptyset$;

RETURN $\mathcal{H}_v$;
END;
```

Algorithm 1.4: *Identifying while Locating*

1.4.1 Hypothesis Generation

The goal of the hypothesis generation process is to use the smallest set of observations that allow to determine the object location. Normally, two independent observations are necessary to determine the object location (see algorithm 1.4). In order to obtain a precise estimation of the object location, functions `select_first_observation` and `select_second_observation` should choose observations that are independent and with low location uncertainty. These observations should also have as few pairings as possible, so that the number of alternative hypotheses be small. We generate all possible pairings for this limited set of observations with the `search_for_pairings` function presented in algorithm 1.1.

1.4.2 Hypothesis Verification

For the verification of the resulting hypotheses, the estimated location of the object allows the system to validate each additional pairing for *global* consistency using the *rigidity* constraint. This constraint validation simply consists in determining whether the location of the observation in the scene coincides with the predicted location of the model feature according to the

```
FUNCTION verify_hypothesis ($h, \mathcal{E}, \mathcal{M}$)

; $h$ : object-location hypothesis to verify
; $\mathcal{E}$: set of available observations
; $\mathcal{M}$: set of model features

; selects an observation and determines which model features satisfy
; the rigidity constraint with it, those pairings are included in the hypothesis' support
; and the object location is refined

  $\mathcal{H} := \emptyset$;
  IF $\mathcal{E} = \emptyset$ THEN
    $\mathcal{H} := \mathcal{H} \cup \{h\}$;
  ELSE
    $e :=$ select_observation ($\mathcal{E}$);                                  (Chapter 6)
    FOR $m \in \mathcal{M}$ DO
      $p := (e, m)$;
      IF satisfy_rigidity_constraint ($\mathbf{L}_h$, $p$) THEN                  (Chapter 3)
        $\mathbf{L}_{h_p} :=$ refine_object_location ($\mathbf{L}_h$, $p$);        (Chapter 2)
        $\mathcal{S}_{h_p} := \mathcal{S}_h \cup \{p\}$;
        $\mathcal{H} := \mathcal{H} \cup$ verify_hypothesis ($h_p$, $\mathcal{E} \setminus \{e\}$ , $\mathcal{M}$);
      FI;
    OD;
    $\mathcal{H} := \mathcal{H} \cup$ verify_hypothesis ($h$, $\mathcal{E} \setminus \{e\}$ , $\mathcal{M}$);
  FI;

RETURN $\mathcal{H}$;
END;
```

Algorithm 1.5: *Data-driven Hypothesis Verification*

hypothesis. Nevertheless, this constraint validation is more discriminant because it validates *global* consistency.

Hypothesis verification can be carried out in a *data-driven* fashion (see algorithm 1.5). An observation is selected, and it is determined whether it can be paired with one of the model features. If an acceptable pairing is found, the function recurs with an object location estimation refined by the inclusion of the new pairing. This process continues until there are no more available observations. Note that only one validation of the rigidity constraint is necessary for each potential pairing. Thus, the number of validations for a given hypothesis grows *linearly* with the number of pairings.

Hypothesis verification can also be carried out in a *model-driven* manner (see algorithm 1.6). This consists in selecting a model feature, predicting its location in the scene, and determining whether it coincides with some of the available observations. It has the advantage that it allows to determine regions in the scene where the features should be searched for. If there is no more sensorial information available, a sensor can be used to scan this

```
FUNCTION verify_hypothesis ($h, \mathcal{E}, \mathcal{M}$)

; $h$ : object-location hypothesis to verify
; $\mathcal{E}$: set of available observations
; $\mathcal{M}$: set of model features

; Selects a model feature and determines which observations satisfy
; the rigidity constraint with it, those pairings are included in the hypothesis' support
; and the object location is refined

  $\mathcal{H} := \emptyset$;
  IF $\mathcal{E} = \emptyset$ THEN
     $\mathcal{H} := \mathcal{H} \cup \{h\}$;
  ELSE
     $m :=$ select_model_feature ($\mathcal{M}$);                          (Chapter 6)
     FOR $e \in \mathcal{E}$ DO
        $p := (e, m)$;
        IF satisfy_rigidity_constraint ($\mathbf{L}_h$, $p$) THEN          (Chapter 3)
           $\mathcal{S}_h := \mathcal{S}_h \cup \{p\}$;
           $\mathbf{L}_h :=$ refine_object_location ($\mathbf{L}_h$, $p$);  (Chapter 2)
        FI;
        $\mathcal{H} := \mathcal{H} \cup$ verify_hypothesis ($h$, $\mathcal{E}$, $\mathcal{M} \setminus \{m\}$);
     OD;
  FI;

RETURN $\mathcal{H}$;
END;
```

Algorithm 1.6: *Model-Driven Hypothesis Verification*

region to verify the presence of the feature.

1.5 Obtaining more Information from the Scene

The result of both recognition schemes may be a set of globally consistent hypotheses that provide alternative interpretations of the observations. The only way in which we can determine which is correct is to obtain additional sensorial information.

One way of disambiguating between several hypotheses is to determine a sensor location from which the obtained measurement (usually a surface point using proximity) would allow us to determine which of the competing hypotheses is correct. Different techniques are applied to this problem [Grimson 86, Cameron 90, Ellis 92]. The general idea is to determine a sensing direction from which the predicted observations corresponding to each alternative hypothesis are more distant from each other, and thus, more discriminant.

This solution can obtain the correct hypothesis very fast. However, it considers only sensing operations corresponding to obtaining points from the surface, and it does not take into account how uncertain the estimated location of the object may be, and how the sensing operation can improve this estimation. For these reasons, we wish to study the use of *goal-directed* perception strategies, in which the system simultaneously analyzes how to verify an object hypothesis and refine its estimated location.

1.6 Organization of the work

In the preceding sections we have outlined the fundamental issues related to geometric object recognition in multisensor systems. In this work we concentrate our efforts in five fundamental areas:

- The obtention of observation models for different types of sensors and geometric features, where both the characteristics of the geometric feature and the precision of the involved sensor are explicitly taken into account. In chapter 2, observation models are developed for two types of sensors: mobile proximity and mobile 2D vision. These observation models will be used as case studies throughout this work, illustrating the generality of our contributions to the object recognition problem.

- The analysis of validation mechanisms, known as *geometric constraints*. In chapter 3 we propose a general procedure to validate both location independent and location dependent constraints under uncertainty. Such procedures take into account the precision of the sensor that gives the observation and the characteristics of the involved geometric features.

- The definition of mechanisms to determine the precision and relevance of an object-location hypothesis, as a function of the set of observations that support the hypothesis. This subject, studied in chapter 4, allows us to propose strategies for the selection of sensorial observations and model features to which direct the system's attention, so that the most plausible hypotheses are generated first and verified rapidly.

- The analysis of perception tasks that can make use of *a priori* information about the location of a feature, with the goal of verifying the presence of the feature in the scene, and reducing the uncertainty of the object location. We propose a method that allows to compare the potential benefit of using different sensors for a perception task, and to compute a sensor location such that this benefit is maximal. These *goal-directed perception strategies* are described in chapter 5.

- The study of the two alternative recognition schemes, the *identifying before locating* and the *identifying while locating* schemes. In chapter 6 we show that from the constraint validation point of view, simultaneous identification and localization can improve the performance of recognition. Given the importance of having a precise estimation of the object location as soon as possible, we develop a general procedure to determine whether a set of observations allow to determine the location of the object, and to calculate this location. Within the *identifying while locating* scheme, we propose a set of strategies for the generation and verification of hypotheses that can help reduce the complexity of the recognition process.

Chapter 2

Modeling Uncertain Geometric Information

Summary

In this chapter we discuss the advantages of the use of probabilistic models over set-based models in representing geometric uncertainty. We favor the use of probabilistic models because of the high complexity associated with the use of set-based models in the representation of the uncertainty of the location of a geometric entity, and because is seems more feasible that sensorial observations of geometric entities have Gaussian distributions of error. We briefly describe the Symmetries and Perturbation model *(SPmodel), a general method for the representation of the location of any geometric entity and its uncertainty, proposed in [Tardós 91]. The SPmodel is a probabilistic model whose main advantage is its generality: it is valid for any object, geometric feature or sensorial observation. It has been used to establish a general integration method that allows to obtain a* suboptimal estimation of location *for objects or features from a set of partial and uncertain sensorial observations [Tardós 91, Tardós 92a, Tardós 92b]. In this chapter we also obtain observation models for different types of geometric elements given by two types of sensors: mobile proximity and mobile 2D vision. These observation models will be used throughout this work as case studies to illustrate the generality of the proposed recognition scheme.*

2.1 Introduction. Set-based .vs. Probabilistic Models

Most of the existing models to represent uncertain geometric information can be classified into two main groups: *set-based models* and *probabilistic models*. In set-based models, the imprecision in the location $\mathbf{x}$ of a geometric entity observed using a sensor P is described by a *bounded region* R_P, which corresponds to the *set of feasible locations* for the geometric entity:

$$\mathbf{x} \in R_P$$

The fusion of geometric information is accomplished using algebraic operations on the regions limited by the error bounds. Given a set of observations $\{R_{P_1}, \ldots, R_{P_n}\}$ of the geometric element, its estimated location is given by (fig. 2.1.a):

$$\overline{R} = \bigcap_{i=1}^{n} R_{P_i}$$

An example of the use of this approach appears in [Brooks 82, Brooks 85], where the uncertain location of a mobile robot is represented with a cylindrical volume of error, associated to the robot location parameters $(x,\ y,\ \theta)$ (in configuration space). Given a set of robot motions, the resulting uncertainty in the new robot location is estimated from the uncertainty associated to each of the commanded robot motions, and from the performed perception operations.

This approach is also used in [Grimson 84, Grimson 90a] to develop a set of geometric constraints to verify the consistency between model and observed features of an object considering the uncertainty associated to the sensor information. In 3D, the uncertainty in position is represented by a spherical region, characterized by its radius, and uncertainty in orientation is represented by a conic region, characterized by its spreading angle. This radius and spreading angle depend on the sensor precision. The work of Ellis [Ellis 91] deals with obtaining tighter error bounds than those obtained by Grimson. Faugeras and Herbert [Faugeras 86] also use tolerances in validating the pairing of sensed and model features.

Probabilistic models represent the uncertain location of a geometric element using a *probability distribution*—usually Gaussian. For Gaussian distributions, a significance value α defines an ellipsoidal region where the probability of finding the true value of $\mathbf{x}$ is $1 - \alpha$, given by:

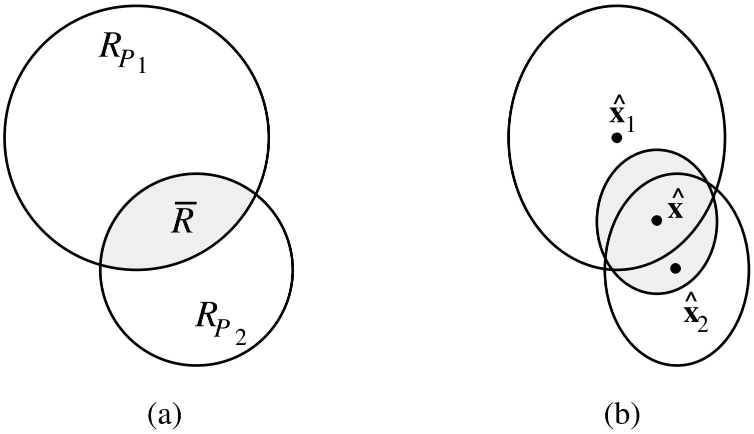


Figure 2.1: (a) Set-based representation and integration; (b) Probabilistic representation and integration

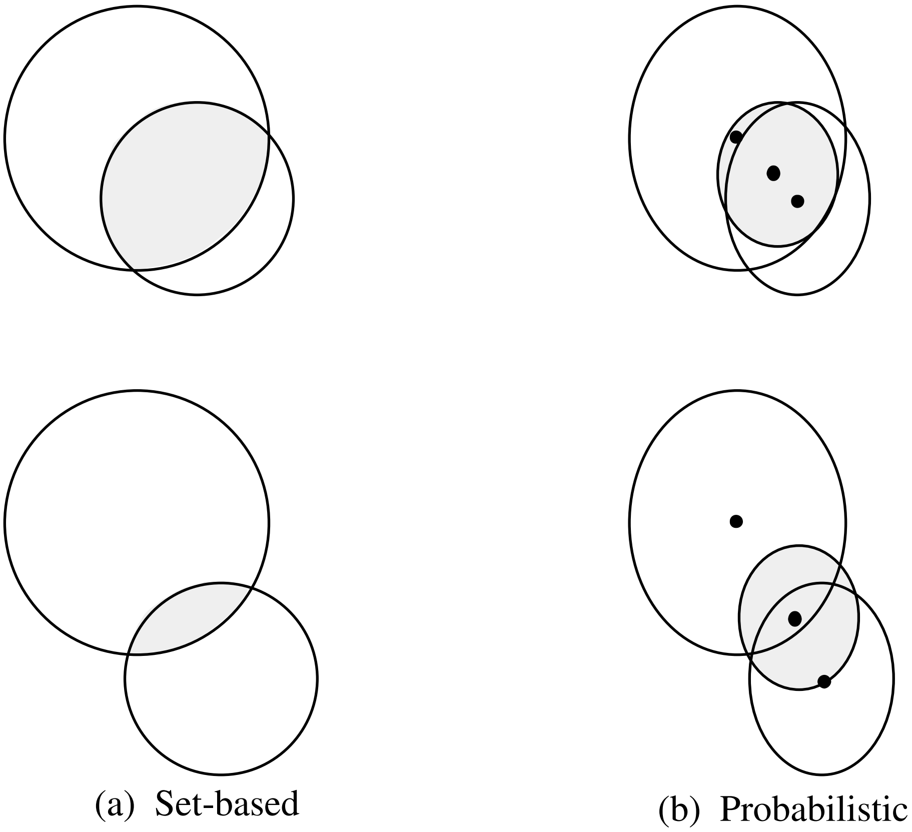
$$P\left\{(\mathbf{x}-\hat{\mathbf{x}})^T C^{-1}(\mathbf{x}-\hat{\mathbf{x}}) < D^2_{m,\alpha}\right\} = 1-\alpha$$

where $\hat{\mathbf{x}}$ is the mean value of $\mathbf{x}$, C its covariance, and $D^2_{m,\alpha}$ is the value of the chi-square distribution with $m = dim(\mathbf{x})$ degrees of freedom, for a significance level α. This allows to represent graphically the uncertainty regions associated with probabilistic models (fig. 2.1.b).

Fusion can be carried out using optimal estimation methods. Given n sensorial observations $\{(\hat{\mathbf{x}}_{P_1}, C_{P_1}), \ldots, (\hat{\mathbf{x}}_{P_n}, C_{P_n})\}$ of the same geometric element, the estimated value and covariance of its location are given by:

$$\hat{\mathbf{x}} = \left(\sum_{i=1}^{n} C_{P_i}^{-1}\right)^{-1} \sum_{j=1}^{n} C_{P_j}^{-1}\,\hat{\mathbf{x}}_{P_j} \quad ; \quad C = \left(\sum_{i=1}^{n} C_{P_i}^{-1}\right)^{-1}$$

The mobile robot HILARE [Chatila 85] uses this approach to estimate its location and model its indoor environment. In [Smith 88], this approach is used to build and update a *stochastic map* of the relationships among a set of geometric entities. The estimation of these relationships is done using the extended Kalman filter (EKF). A Bayesian approach to obtaining a consistent interpretation of a set of uncertain and disparate sensor observations



(a) Set-based

(b) Probabilistic

Figure 2.2: (a) More discrepant observations produce more precise estimations of uncertainty; (b) The discrepancy of the integrated observations affects only the mean value of the estimation

is proposed in [Durrant-Whyte 87]. Under the Gaussianity hypothesis, this procedure is equivalent to the Kalman filter. The 3DPO system [Bolles 86] uses this representation model to test the consistency of matches between sensed and model features of objects. This representation model is also used in [Porrill 88] and [Pollard 89] to obtain estimated values and covariances for the geometric relations between uncertain observations.

There is considerable controversy about the use of either model [Smith 88, Sabater 91, Hager 93]. With respect to this controversy, there are two fundamental issues that must be considered: *appropriateness* and *practicality.*

2.1.1 Appropriateness

Which fusion method models more adequately the problem is an important matter because each fusion method leads to very different estimations of uncertainty:

- In set-based methods, more discrepant observations lead to a greater reduction of uncertainty in the location of the involved geometric element (fig. 2.2.a), while in probabilistic methods the resulting uncertainty is only function of the uncertainty of the observations, not of the observation values (fig. 2.2.b). This is because set-based methods rely on the certainty that the true location lies within the given bound. Thus, this error bound must be large enough to assure this—which can lead to very conservative estimations of error.

- Fusion methods in probabilistic models rely on the independence of the fused observations. The fusion of two independent observations with similar value leads to uncertainty reduction, which is not the case in set-based methods (fig. 2.3). Thus, if the observations are not independent, probabilistic methods lead to optimistic estimations of uncertainty [Sabater 91]. However, independence can be guaranteed using different sensors, or different points of view with a mobile sensor.

- From a probabilistic point of view, set-based models can be thought of as having *uniform* distribution functions over the region R_P. It seems less feasible that sensorial observations of geometric features have uniform distributions of error. Theoretically, the central limit theorem states that the combination of several independent sources of information used in estimating a location tends to follow a Gaussian distribution [Smith 88, Durrant-Whyte 88]. Some experiments carried

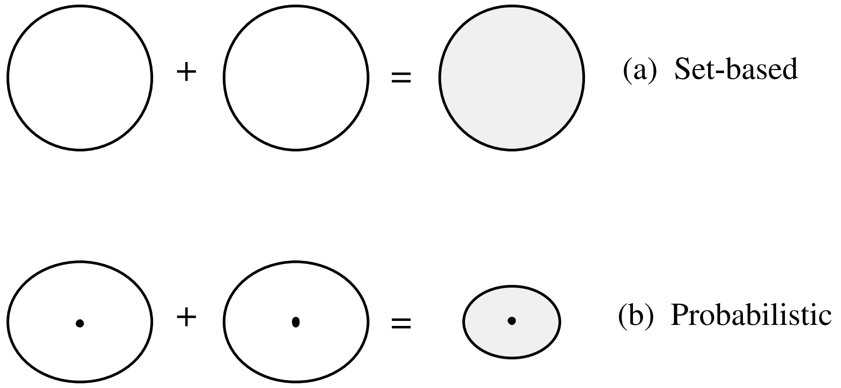


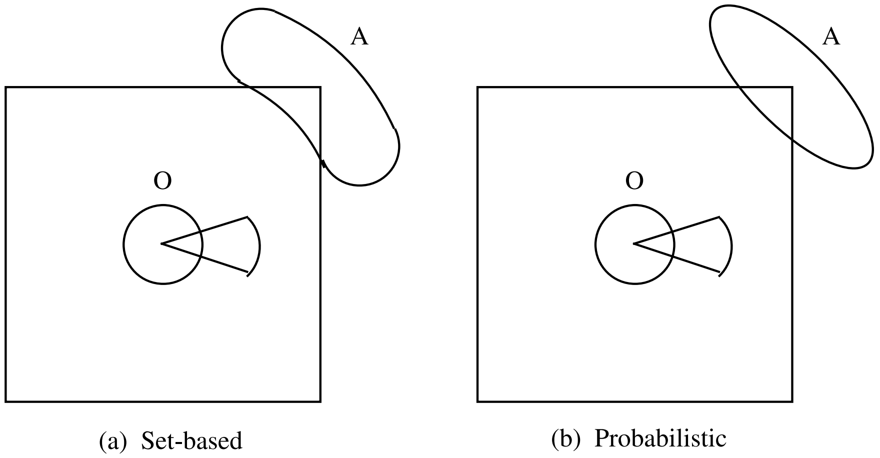
Figure 2.3: The fusion of two similar observations observation (a) does not affect the result of the estimation; (b) reduces the uncertainty of the estimation

out in our laboratory with proximity and vision sensors also tend to support the gaussianity hypothesis [Herranz 91, Sagüés 92a].

2.1.2 Practicality

There are two aspects to consider when analyzing the complexity of managing uncertain geometric information:

- *Complexity in propagating uncertainty:* the complexity of set-based representations in nonlinear problems can be high [Hager 93]. Localization problems are nonlinear problems involving position and orientation components, where these are *coupled*. Consider the object in figure 2.4.a. Given the location of the object represented by the location of a reference O in its center, with uncertainty in position and orientation represented graphically by a circle and a cone respectively, the resulting uncertainty in the location of point A is complex to represent. If we consider problems in 3D space, the complexity of representation becomes unmanageable. Probabilistic methods provide simple linear approximations to represent the uncertainty associated to the location of A (fig. 2.4.b). Supposing we are dealing with small amounts of error, the approximation is precise enough.



(a) Set-based

(b) Probabilistic

Figure 2.4: In representing localization problems (a) set-based methods are very complex; (b) probabilistic methods provide simple approximations

- *Complexity in fusing:* in set-based methods, fusion is straight-forward only in the one-dimensional linear case. The complexity of the fusion algorithm grows with the number of parameters to be estimated (an important parallelization effort to reduce the required computation time for carrying out set-based operations can be found in [Fisher 91]). Some simplifications can be considered to reduce this complexity, but at the cost of loosing precision in the estimation (fig. 2.5). An example of this approach to complexity reduction can be found in [Brooks 85].

In summary, since we are dealing with a nonlinear problem involving the estimation of six parameters, set-based methods will be either much more complex to apply, or the resulting estimations of errors will be more conservative.

2.2 The SPmodel

The arguments given above have led us to prefer the use of probabilistic models for the representation of uncertain geometric information. In probabilistic uncertainty representation models, the location of an element is represented by a parameter vector, and the available knowledge about it is

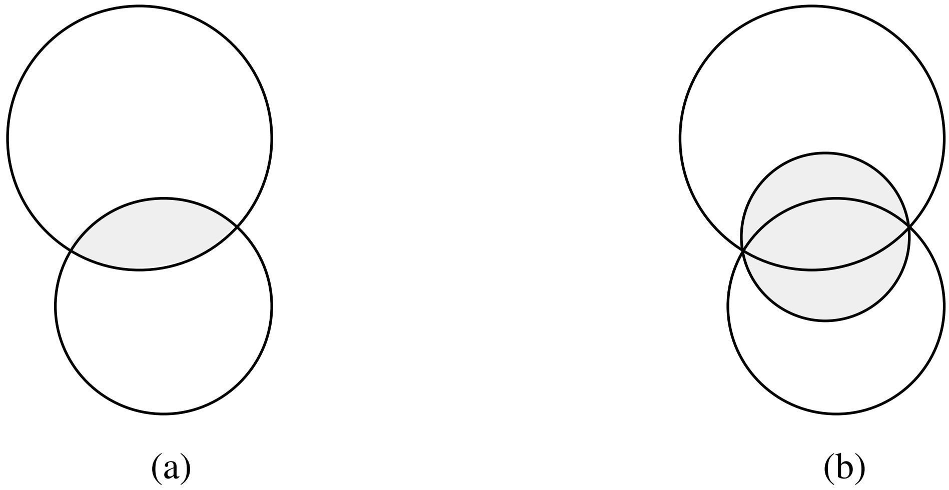


Figure 2.5: When fusing information using set-based methods, (a) the form of the resulting estimation is increasingly more complex; or (b) an artificial error must be introduced to maintain the simplicity of form of the estimation

characterized by the mean and covariance of an associated probability distribution function. The fusion of sensorial information can be done using optimal estimation techniques, such as the extended Kalman filter.

The main drawback of current approaches is that they use a different set of parameters to represent the location of each type of geometric element. In [Tardós 91, Tardós 92a] Tardós presented a general method for the representation of the location of any geometric entity and its uncertainty: the Symmetries and Perturbations model (SPmodel). For completeness, and to fix the notation used, this model and the associated fusion mechanism are briefly presented next.

The SPmodel combines the use of probability theory to represent the *imprecision* in the location of a geometric element, and the theory of symmetries to represent the *partiality* due to the characteristics of each type of geometric element. In [Popplestone 84], Popplestone points out the benefit of applying group theory [Hall 76]—on which the theory of symmetries is based—to analyze some types of relationships between features of an object. Thomas [Thomas 88b] calculates the common symmetries of a set of geometric elements, and uses them to determine assembly plans satisfying the constraints imposed by each geometric element.

In the SPmodel, a reference E is associated to every geometric element e. Its location is given by the transformation t_{WE} relative to a base reference W. To represent this transformation, we use a *location vector* $\mathbf{x}_{WE}$, composed of three Cartesian coordinates and three Roll-Pitch-Yaw angles:

$$\mathbf{x}_{WE} = (x,\ y,\ z,\ \psi,\ \theta,\ \phi)^T$$

where:

$$t_{WE} = \mathrm{Trans}(x,\ y,\ z) \cdot \mathrm{Rot}(z,\ \phi) \cdot \mathrm{Rot}(y,\ \theta) \cdot \mathrm{Rot}(x,\ \psi)$$

The composition of location vectors is represented with operator $\oplus$, and the composition with the inverse is abbreviated as $\ominus$:

$$\begin{array}{l} \mathbf{x}_{AC} = \mathbf{x}_{AB} \oplus \mathbf{x}_{BC} = \mathbf{x}_{AB} \ominus \mathbf{x}_{CB} \\ \mathbf{x}_{CB} = \ominus \mathbf{x}_{BC} \end{array}$$

The estimation of the location of an element is denoted by $\hat{\mathbf{x}}_{WE}$, and the estimation error is represented locally by a *differential location vector* $\mathbf{d}_E$ relative to the reference attached to the element. Thus, the true location of the element is:

$$\mathbf{x}_{WE} = \hat{\mathbf{x}}_{WE} \oplus \mathbf{d}_E$$

Our model also exploits the concept of *symmetries* of a geometric element, defined as the set $\mathcal{S}_E$ of transformations that preserve the element. It has been shown that the symmetries of any geometric element are a subgroup of the group of transformation $(\mathcal{T}, \cdot)$. For example, the symmetries of an infinite edge are the set of continuous translations (T_x) and rotations (R_x) along the edge. There is also a cyclic symmetry of 180 degrees around any axis perpendicular to the edge, corresponding to the two opposite edge orientations (fig. 2.6).

Cyclic symmetries must be taken into account as alternate hypotheses when matching two features in the recognition process. To account for the continuous motion symmetries, we assign in $\mathbf{d}_E$ a null value to the degrees of freedom corresponding to them, because they do not represent an effective location error. We call *perturbation vector* the vector $\mathbf{p}_E$ formed by the non null elements of $\mathbf{d}_E$. Both vectors can be related by a row selection matrix B_E that we call *self-binding matrix* of the geometric element:

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

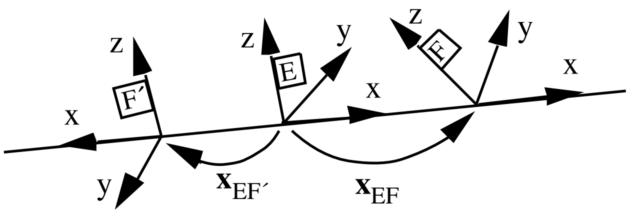


Figure 2.6: Examples of symmetries of an infinite edge

For example, in the case of an edge, the symmetries are $T_x\, R_x$, and thus we have:

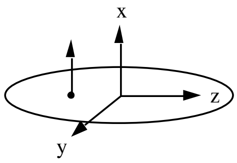
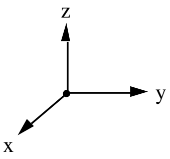
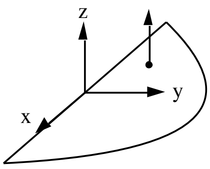
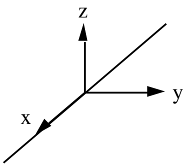
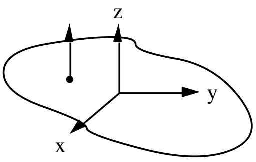
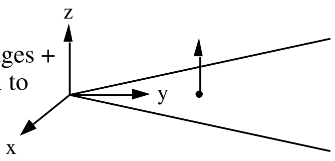
$$\begin{aligned}\mathbf{d}_E &= (0,\ dy,\ dz,\ 0,\ d\theta,\ d\phi)^T\\ \mathbf{p}_E &= (dy,\ dz,\ d\theta,\ d\phi)^T\\ B_E &= \begin{bmatrix}0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}\end{aligned}$$

To obtain the self-binding matrix B_E, it is enough to eliminate from the unity matrix I_6 the rows corresponding to the symmetries of the geometric element (the first for T_x and the fourth for R_x, in the case of an edge). Based on these ideas, the SPmodel represents the information about the location of a geometric element e by a triplet $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E, C_E)$, where:

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

Transformation $\hat{\mathbf{x}}_{WE}$ is an estimation taken as base for perturbations, $\hat{\mathbf{p}}_E$ is the estimated value of the perturbation vector, and C_E its covariance. When $\hat{\mathbf{p}}_E = 0$, we say that the estimation is *centered.* Figure 2.7 shows some examples of geometric elements with their continuous motion symmetries.

The main advantage of this model is its generality: it is valid for any object, geometric feature or sensorial observation. Moreover, the representation of uncertainty using a perturbation vector does not depend on the base reference used, has a clear interpretation, and is not overparameterized. Problems related to singularities found in other representations are also avoided, because the perturbation vector is considered small, that is,

Symmetries	**Example**	**Other Examples**
Rx	Circle 	Conical Surface Edge with extremes
Rxyz	Vertex 	Spherical surface
Tx	Semidihedral (edge + normal to plane) 	Dihedral (edge + two normals)
Tx Rx	Infinite Edge 	Infinite Cylindrical Surface
Txy Rz	Infinite Plane 	2D Edge (Vision)
$\mathbb{1}$	Corner (two edges + normal to plane) 	Coplanar Edges (non parallel)

$$T_x = \{\text{Trans}(a,0,0) \mid a \in I\!\!R\}$$
$$T_{xyz} = \{\text{Trans}(a,b,c) \mid a,b,c \in I\!\!R\}$$
$$R_x = \{\text{Rot}(x,\psi) \mid \psi \in (-\pi,\ \pi]\}$$
$$R_{xyz} = \{\text{Rot}(\mathbf{u},\theta) \mid \mathbf{u} \in I\!\!R^3, \theta \in (-\pi,\pi]\}$$

Figure 2.7: Continuous motion symmetries of some geometric elements

its values will be far from the singularity of the RPY angles representation, $\theta = \pm\pi/2$. In the next subsections we will define the fundamental operations derived from the SPmodel.

2.2.1 Operations with the SPmodel

In most operations with uncertain locations represented with the SPmodel, it is necessary to calculate the uncertainty in one reference due to the uncertainty in the location of another reference. The expressions for transforming differential locations between references are:

$$\begin{array}{l} \mathbf{d}_A \oplus \mathbf{x}_{AB} = \mathbf{x}_{AB} \oplus \mathbf{d}_B \\ \mathbf{d}_B = J_{AB}^{-1}\,\mathbf{d}_A = J_{BA}\,\mathbf{d}_A \\ Cov(\mathbf{d}_B) = J_{BA}\,Cov(\mathbf{d}_A)\,J_{BA}^T \end{array} \tag{2.1}$$

where J_{BA} is the Jacobian of transformation $\mathbf{x}_{BA}$ [Paul 81] (see appendix A).

Centering an Uncertain Location

Given an uncertain location $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E, C_E)$, where $\hat{\mathbf{p}}_E \neq 0$, it can be transformed to $\mathbf{L}_{WE'} = (\hat{\mathbf{x}}_{WE'}, \hat{\mathbf{p}}_{E'}, C_{E'})$, where $\hat{\mathbf{p}}_{E'} = 0$, as follows:

$$\begin{array}{rcl} \hat{\mathbf{x}}_{WE'} & = & \hat{\mathbf{x}}_{WE} \oplus B_E^T \hat{\mathbf{p}}_E \\ C_{E'} & = & \left(B_E J_{2\oplus}^{-1}\left\{B_E^T\hat{\mathbf{p}}_E, 0\right\} B_E^T\right)\, C_E\, \left(B_E J_{2\oplus}^{-1}\left\{B_E^T\hat{\mathbf{p}}_E, 0\right\} B_E^T\right)^T \end{array} \tag{2.2}$$

where $J_{2\oplus}$ is the Jacobian of the composition of location vectors [Smith 88] (see appendix A):

$$J_{2\oplus}\{\mathbf{x}_1, \mathbf{x}_2\} = \left.\frac{\partial(\mathbf{y} \oplus \mathbf{z})}{\partial \mathbf{z}}\right|_{\mathbf{y}=\mathbf{x}_1,\, \mathbf{z}=\mathbf{x}_2}$$

Changing the Associated Reference

Consider a geometric element e, whose estimated location is given by $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E, C_E)$. Suppose we choose to represent the location of the geometric element using a different associated reference F. Let $\mathbf{x}_{EF}$ represent the relative location between E and F. Vector $\mathbf{x}_{WF}$ will be given by:

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

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

$$\begin{aligned}
\hat{\mathbf{x}}_{WF} &= \hat{\mathbf{x}}_{WE} \oplus \mathbf{x}_{EF} \\
\hat{\mathbf{p}}_F &= B_F \, J_{FE} \, B_E^T \, \hat{\mathbf{p}}_E \\
C_F &= B_F J_{FE} B_E^T \; C_E \; B_E J_{FE}^T B_F^T \qquad (2.3)
\end{aligned}$$

This operation, the composition between an uncertain location $\mathbf{L}_{WE}$ and a location vector $\mathbf{x}_{EF}$, will also be denoted by $\oplus$. Thus:

$$\mathbf{L}_{WF} = \mathbf{L}_{WE} \oplus \mathbf{x}_{EF}$$

where $\mathbf{L}_{WF}$ is given by (2.3). This result is useful in situations where reference F is more appropriate to express the location of the geometric element. Such a situation occurs in the computation of the geometric relations between two uncertain features (chapter 3).

Composing two Uncertain Locations

Consider a geometric element e, whose estimated location with respect to a reference F is given by $\mathbf{L}_{FE} = (\hat{\mathbf{x}}_{FE}, \hat{\mathbf{p}}_E, C_E)$. Suppose reference F is affected by a location error with respect to W, *independent* from the error which affects E, and this error is given by $\mathbf{L}_{WF} = (\hat{\mathbf{x}}_{WF}, \hat{\mathbf{p}}_F, C_F)$. To compute the uncertain location of E with respect to W we proceed as follows: vector $\mathbf{x}_{WE}$ will be given by:

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

where J_{EF} is the Jacobian of transformation $\mathbf{x}_{EF}$. Given that the values of the components of differential location vectors can be considered small, their composition ($\oplus$) can be approximated by their sum ($+$). Thus, the uncertain location of E with respect to W is given by $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E^W, C_E^W)$, where:

$$\begin{aligned} \hat{\mathbf{x}}_{WE} &= \hat{\mathbf{x}}_{WF} \oplus \hat{\mathbf{x}}_{FE} \\ \hat{\mathbf{p}}_E^W &= B_E J_{EF} B_F^T \hat{\mathbf{p}}_F + \hat{\mathbf{p}}_E \\ C_E^W &= B_E J_{EF} B_F^T C_F B_F J_{EF}^T B_E^T + C_E \end{aligned} \tag{2.4}$$

Again, we will use the symbol $\oplus$ to express the composition between uncertain locations. Thus:

$$\mathbf{L}_{WE} = \mathbf{L}_{WF} \oplus \mathbf{L}_{FE}$$

where $\mathbf{L}_{WE}$ is given by (2.4). This composition is useful in situations where the location of a geometric element is expressed with respect to a reference affected by an error, such as an observation obtained by a sensor mounted on the robot hand (section 2.4.1).

Changing the Base Reference

Consider a geometric element e, whose estimated location with respect to a reference F is given by $\mathbf{L}_{FE} = (\hat{\mathbf{x}}_{FE}, \hat{\mathbf{p}}_E, C_E)$. Suppose we wish to express the location of E with respect to an alternative reference W. Vector $\mathbf{x}_{WE}$ is given by:

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

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

$$\hat{\mathbf{x}}_{WE} = \mathbf{x}_{WF} \oplus \hat{\mathbf{x}}_{FE} \tag{2.5}$$

Again, we will use the symbol $\oplus$ to express the composition between a location vector and an uncertain location. Thus:

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

where $\mathbf{L}_{WE}$ is given by (2.5).

2.3 Integrating Uncertain Geometric Information

In [Tardós 91], 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* or the *extended information filter* [Gelb 74, Bozzo 83]. In this work we prefer the information filter formulation because it simplifies the analysis of the influence of each observation on the estimation of the object location.

The extended information filter is formulated 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 $\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) = 0$. Since $\mathbf{f}_k$ is nonlinear due to orientation terms, we use a first order approximation:

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

where:

$$\mathbf{h}_k = \mathbf{f}_k(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k) \;\; ; \;\; H_k = \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k)} \;\; ; \;\; G_k = \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{y}}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k)} \tag{2.6}$$

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

$$\hat{\mathbf{x}}_n = P_n M_n \;\; ; \;\; P_n^{-1} = Q_n = \sum_{k=1}^{n} F_k \;\; ; \;\; M_n = -\sum_{k=1}^{n} N_k \tag{2.7}$$

where:

$$F_k = H_k^T (G_k S_k G_k^T)^{-1} H_k \;\; ; \;\; N_k = H_k^T (G_k S_k G_k^T)^{-1} h_k \tag{2.8}$$

This is the nonrecursive 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 [Tardós 91]. We apply the information filter to three different estimation problems, as explained in the next subsections.

2.3.1 Feature Location Estimation from a Set of Subfeatures

Partial observations of a feature, such as a point on an edge or on a plane, the normal of a plane, or a 2D image of an edge (we denominate them *subfeatures*), can be used to estimate the location of the associated geometric feature [Tardós 92a]. Let $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E, C_E)$ represent the estimated location of the feature e. Let P_k be a reference frame associated to a partial observation p_k, whose location is represented by $\mathbf{L}_{WP_k} = (\hat{\mathbf{x}}_{WP_k}, \hat{\mathbf{p}}_{P_k}, C_{P_k})$. We can apply the information filter to this problem by considering the perturbation vector of the feature as the state to be estimated, and the perturbation vector of each subfeature as the measurements (for simplicity, and without loss of generality, we will consider both P and E as *centered* estimations):

$$\begin{array}{rcl} \mathbf{x} = \mathbf{p}_E & ; & \hat{\mathbf{x}} = \hat{\mathbf{p}}_E = 0 \\ \mathbf{y}_k = \mathbf{p}_{P_k} & ; & \hat{\mathbf{y}}_k = \hat{\mathbf{p}}_{P_k} = 0 \\ S_k & = & C_{P_k} \end{array}$$

Every subfeature imposes a *constraint* on the location of the associated feature. The pairing between the observation and the feature imposes a constraint on the relative transformation $\mathbf{x}_{P_kE}$ that can be expressed by means of the *binding matrix of the pairing* B_{P_kE}:

$$B_{P_kE}\, \mathbf{x}_{P_kE} = 0$$

This constraint is denominated the *inverse constraint*, and it expresses the fact that the observed subfeature location and the location of its associated feature must coincide, up to the symmetries of the pairing. It is also possible to express the pairing using the *direct constraint*:

$$B_{EP_k}\, \mathbf{x}_{EP_k} = 0$$

These constraints give us the implicit measurement function $\mathbf{f}_k$. Whether we can use the direct or inverse constraint depends on the types of the involved subfeature and feature. In case we use the *direct constraint* as the implicit state function, we have (fig. 2.8):

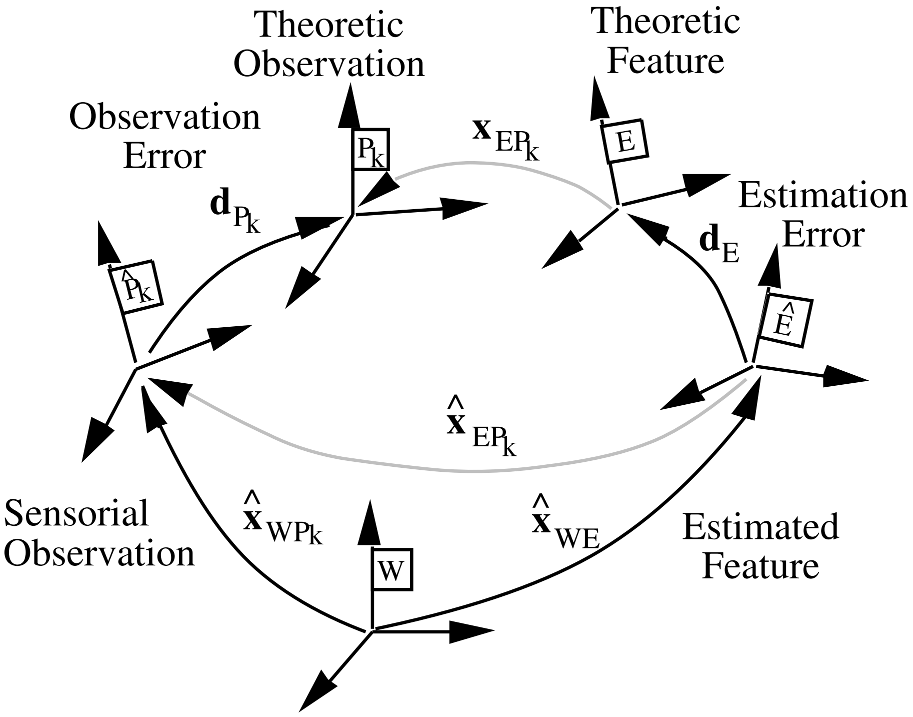


Figure 2.8: Transformations involved in the pairing between an observed subfeature P_k and its associated feature E (from [Tardós 91])

$$\begin{aligned}
\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) &= B_{EP_k}\mathbf{x}_{EP_k} \\
&= B_{EP_k}(\ominus B_E^T \mathbf{p}_E \oplus \hat{\mathbf{x}}_{EP_k} \oplus B_{P_k}^T \mathbf{p}_{P_k}) \\
&= B_{EP_k}(\ominus B_E^T \mathbf{x} \oplus \hat{\mathbf{x}}_{EP_k} \oplus B_{P_k}^T \mathbf{y}_k) \\
&= 0
\end{aligned}$$

Since we consider that the feature and the subfeatures are centered estimations, from (2.6) we have:

$$\begin{aligned}
\mathbf{h}_k &= B_{EP_k}\hat{\mathbf{x}}_{EP_k} \\
H_k &= -B_{EP_k} J_{1\oplus}\{0, \hat{\mathbf{x}}_{EP_k}\} B_E^T \\
G_k &= B_{EP_k} J_{2\oplus}\{\hat{\mathbf{x}}_{EP_k}, 0\} B_{P_k}^T
\end{aligned} \tag{2.9}$$

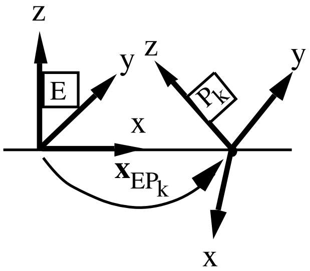


Figure 2.9: The relative location between an edge and a point on the edge is given by $\mathbf{x}_{EP_k} = (x,\, 0,\, 0,\, \psi,\, \theta,\, \phi)^T$

where $J_{1\oplus}$ and $J_{2\oplus}$ are the Jacobians of the composition of location vectors [Smith 88] (see appendix A):

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

In case we use the *inverse constraint* as the implicit state function, we have:

$$\begin{aligned} \mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) &= B_{P_kE}\mathbf{x}_{P_kE} \\ &= B_{P_kE}(\ominus B_{P_k}^T\mathbf{p}_{P_k} \oplus \hat{\mathbf{x}}_{P_kE} \oplus B_E^T\mathbf{p}_E) \\ &= B_{P_kE}(\ominus B_{P_k}^T\mathbf{y}_k \oplus \hat{\mathbf{x}}_{P_kE} \oplus B_E^T\mathbf{x}) \\ &= 0 \end{aligned}$$

Once more, considering that the feature and the subfeatures are centered estimations, from (2.6) we have:

$$\begin{aligned} \mathbf{h}_k &= B_{P_kE}\hat{\mathbf{x}}_{P_kE} \\ H_k &= B_{P_kE}J_{2\oplus}\{\hat{\mathbf{x}}_{P_kE}, 0\}B_E^T \\ G_k &= -B_{P_kE}J_{1\oplus}\{0, \hat{\mathbf{x}}_{P_kE}\}B_{P_k}^T \end{aligned} \tag{2.10}$$

Example 2.1: *Estimating the location of an edge*
A problem of this type is the estimation of the location of an edge from a set of

observations of points on the edge (fig. 2.9). In this case, the pairing relationship can be expressed using the inverse constraint. Given that the point belongs to the edge, the set of possible values for the transformation between the edge E and the point P_k is given by $T_x\, R_{xyz}$. Thus, the binding matrix of this pairing is given by:

$$B_{EP_k} = \left[\begin{array}{cccccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{array} \right]$$

This binding matrix expresses that if the relative location between an edge and a point is given by the relative location vector $\mathbf{x}_{EP_k} = (x,\, y,\, z,\, \psi,\, \theta,\, phi)^T$, then the point belongs to the edge if:

$$B_{EP_k}\mathbf{x}_{EP_k} = (y,\, z)^T = 0$$

◇

2.3.2 Object Location Estimation from a Set of Features

We can use the information filter to estimate the location of an object from a set of observations of its features [Tardós 92b]. Let $\mathbf{L}_{WO} = (\hat{\mathbf{x}}_{WO}, \hat{\mathbf{d}}_O, C_O)$ represent the estimated location of the object O. Let E_k be a reference frame associated to an observed feature e_k, whose location is represented by $\mathbf{L}_{WE_k} = (\hat{\mathbf{x}}_{WE_k}, \hat{\mathbf{p}}_{E_k}, C_{E_k})$. Let M_k be the reference associated to the model feature m_k, to which we want to pair it. The relative location between the model feature and the object is represented by $\mathbf{x}_{OM_k}$, and can be found in the object model. In this case, we consider the perturbation vector of the object as the state to be estimated, and the perturbation vector of the feature as the measurement:

$$\begin{array}{rcl} \mathbf{x} = \mathbf{d}_O & ; & \hat{\mathbf{x}} = \hat{\mathbf{d}}_O = 0 \\ \mathbf{y}_k = \mathbf{p}_E & ; & \hat{\mathbf{y}}_k = \hat{\mathbf{p}}_E = 0 \\ S_k & = & C_{E_k} \end{array}$$

In this case, the implicit measurement function $\mathbf{f}_k$ corresponds to the pairing between the observation and the model feature, which can be expressed as the inverse constraint on the relative transformation $\mathbf{x}_{E_kM_k}$ by means of the self-binding matrix of the feature B_{E_k}:

$$B_{E_k}\,\mathbf{x}_{E_kM_k} = 0$$

This constraint, denominated *inverse constraint*, expresses the fact that the location of the observed feature and the estimated location of the model feature must coincide up to symmetries. We have (fig. 2.10):

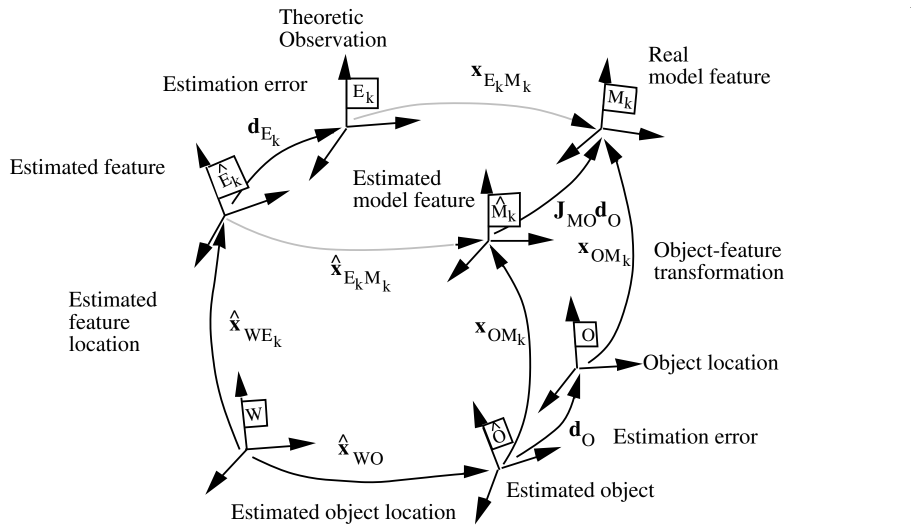


Figure 2.10: Transformations involved in the pairing between an observed feature E_k and its associated model feature M_k (from [Tardós 91])

$$\begin{aligned}
\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) &= B_{E_k}\mathbf{x}_{E_kM_k} \\
&= B_{E_k}(\ominus B_{E_k}^T \mathbf{p}_{E_k} \oplus \hat{\mathbf{x}}_{E_kM_k} \oplus J_{M_kO}\mathbf{d}_O) \\
&= B_{E_k}(\ominus B_{E_k}^T \mathbf{y}_k \oplus \hat{\mathbf{x}}_{E_kM_k} \oplus J_{M_kO}\mathbf{x}) \\
&= 0
\end{aligned}$$

where J_{M_kO} is the Jacobian of transformation $\mathbf{x}_{M_kO}$ between the model feature and the object, which can be obtained from the geometric model of the object. Since we consider that the feature and the object are centered estimations, from (2.6) we have:

$$\begin{aligned}
\mathbf{h}_k &= B_{E_k}\hat{\mathbf{x}}_{E_kM_k} \\
H_k &= B_{E_k}J_{2\oplus}\{\hat{\mathbf{x}}_{E_kM_k}, 0\}B_{E_k}^T B_{E_k} J_{M_kO} \\
G_k &= -B_{E_k}J_{1\oplus}\{0, \hat{\mathbf{x}}_{E_kM_k}\}B_{E_k}^T
\end{aligned} \tag{2.11}$$

Example 2.2: *Estimating the object location using edges*
Given an observed edge with associated reference E_k, and a model edge with refer-

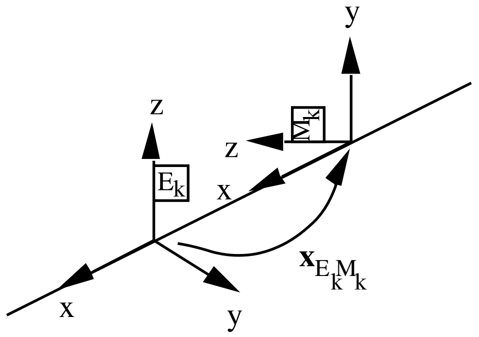


Figure 2.11: The relative location between an observed edge and its corresponding model edge is given by $\mathbf{x}_{E_kM_k} = (x,\, 0,\, 0,\, \psi,\, 0,\, 0)^T$

ence M_k, their coincidence can be expressed using the inverse constraint (fig. 2.11). Given that the self-binding matrix of the edge B_{E_k} is given by:

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

if the relative location between the observed edge is given by the relative location vector $\mathbf{x}_{E_kM_k} = (x,\, y,\, z,\, \psi,\, \theta,\, \phi)^T$, we have that the inverse constraint is:

$$B_{E_k}\mathbf{x}_{E_kM_k} = (y,\, z,\, \theta,\, \phi)^T = 0$$

◇

2.3.3 Object Location Estimation from a Set of Subfeatures

It is also possible to directly integrate partial observations of a feature to improve the estimated location of the object. In this case we apply the information filter in the following way: let P_k be a reference frame associated to the observed subfeature, and let M_k be the one associated to the corresponding model feature. In this case we consider the perturbation vector of the object as the state to be estimated, and the perturbation vector of the subfeature as the measurement:

$$\mathbf{x} = \mathbf{d}_O \quad ; \quad \hat{\mathbf{x}} = \hat{\mathbf{d}}_O = 0$$

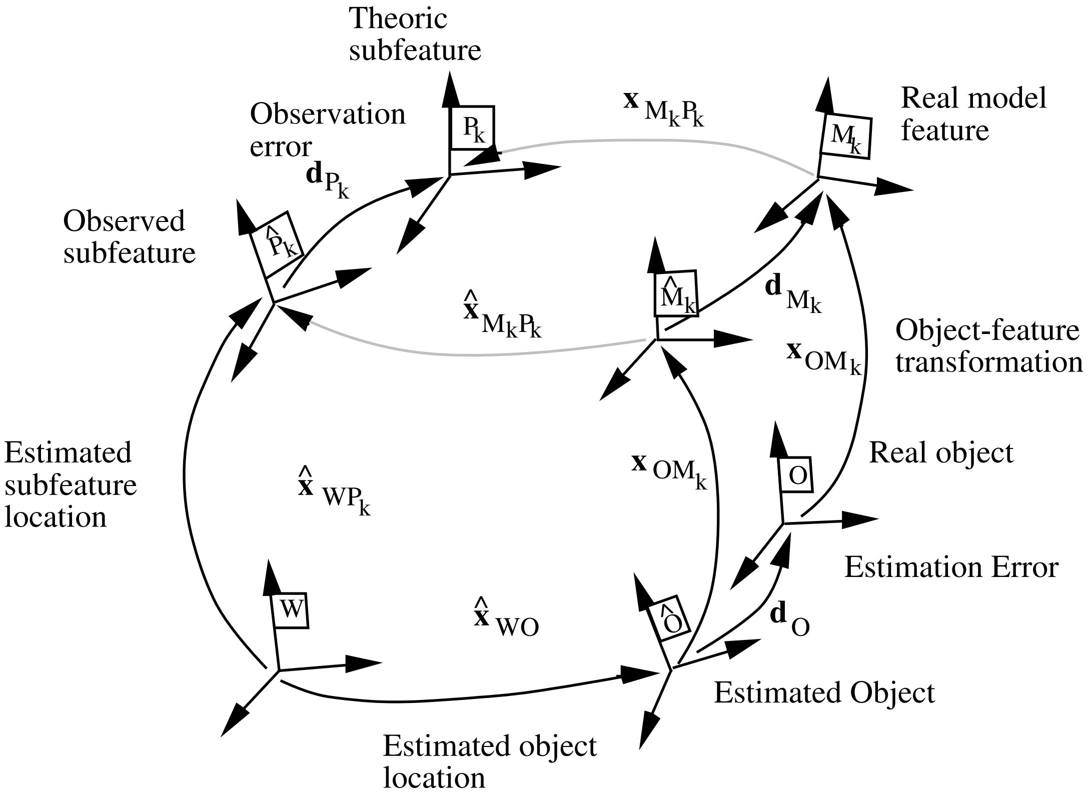


Figure 2.12: Transformations involved in the pairing between an observed subfeature P_k and its associated model feature M_k (from [Tardós 91])

$$\mathbf{y}_k = \mathbf{p}_{P_k} \quad ; \quad \hat{\mathbf{y}}_k = \hat{\mathbf{p}}_{P_k} = 0$$
$$S_k = C_{P_k}$$

The pairing between both can be expressed as a constraint on the relative transformation $\mathbf{x}_{M_kP_k}$ by means of the binding matrix of the pairing. Whether we can use the direct or inverse constraint as implicit measurement function $\mathbf{f}_k$ depends on the type of subfeature and feature:

$$\begin{array}{ll} B_{M_kP_k}\mathbf{x}_{M_kP_k} = 0 & \text{Direct Constraint} \\ B_{P_kM_k}\mathbf{x}_{P_kM_k} = 0 & \text{Inverse Constraint} \end{array}$$

If we use the *direct constraint* we have that (fig. 2.12):

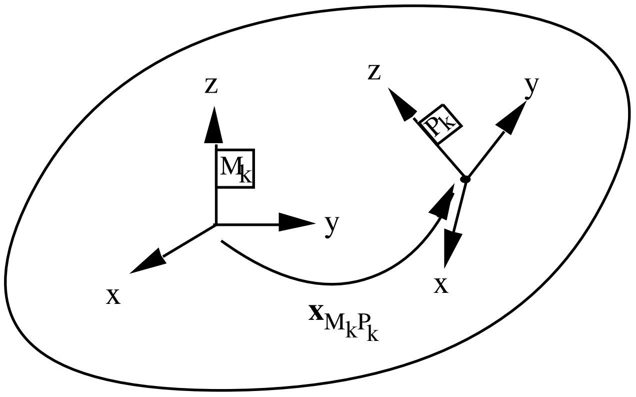


Figure 2.13: The relative location between a plane and a point on the plane is given by $\mathbf{x}_{M_kP_k} = (x,\, y,\, 0,\, \psi,\, \theta,\, \phi)^T$

$$\begin{aligned}
\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) &= B_{M_kP_k}\mathbf{x}_{M_kP_k} \\
&= B_{M_kP_k}(\ominus J_{M_kO}\mathbf{d}_O \oplus \hat{\mathbf{x}}_{M_kP_k} \oplus B_{P_k}^T\mathbf{p}_{P_k}) \\
&= B_{M_kP_k}(\ominus J_{M_kO}\mathbf{x} \oplus \hat{\mathbf{x}}_{M_kP_k} \oplus B_{P_k}^T\mathbf{y}_k) \\
&= 0
\end{aligned}$$

Considering the subfeature and the object location estimations centered, from (2.6) we have:

$$\begin{aligned}
\mathbf{h}_k &= B_{M_kP_k}\hat{\mathbf{x}}_{M_kP_k} \\
H_k &= -B_{M_kP_k}J_{1\oplus}\{0, \hat{\mathbf{x}}_{M_kP_k}\}J_{MO} \\
G_k &= B_{M_kP_k}J_{2\oplus}\{\hat{\mathbf{x}}_{M_kP_k}, 0\}B_{P_k}^T
\end{aligned} \tag{2.12}$$

If we use the *inverse constraint* we have that:

$$\begin{aligned}
\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) &= B_{P_kM_k}\mathbf{x}_{P_kM_k} \\
&= B_{P_kM_k}(\ominus B_{P_k}^T\mathbf{p}_{P_k} \oplus \hat{\mathbf{x}}_{P_kM_k} \oplus J_{M_kO}\mathbf{d}_O) \\
&= B_{P_kM_k}(\ominus B_{P_k}^T\mathbf{y}_k \oplus \hat{\mathbf{x}}_{P_kM_k} \oplus J_{M_kO}\mathbf{x}) = 0
\end{aligned}$$

Considering that the feature and the object are centered estimations we have:

$$\begin{aligned}\mathbf{h}_k &= B_{P_k M_k} \hat{\mathbf{x}}_{P_k M_k} \\ H_k &= B_{P_k M_k} J_{2\oplus}\{\hat{\mathbf{x}}_{P_k M_k}, 0\} J_{M_k O} \\ G_k &= -B_{P_k M_k} J_{1\oplus}\{0, \hat{\mathbf{x}}_{P_k M_k}\} B_{P_k}^T \end{aligned} \tag{2.13}$$

Example 2.3: *Estimating the object location using points on planes*
Given an observation of a point on a plane, with associated reference P_k, and a model plane with reference M_k, their coincidence can be expressed using the direct constraint (fig. 2.13). Given that the point belongs to the plane, the set of possible values for the transformation between the plane M_k and the point P_k is given by $T_{xy}\, R_{xyz}$. Thus, the binding matrix of this pairing is given by:

$$B_{M_k P_k} = \left[\begin{array}{cccccc} 0 & 0 & 1 & 0 & 0 & 0 \end{array} \right]$$

This binding matrix expresses that if the relative location between a plane and a point is given by $\mathbf{x}_{M_k P_k} = (x,\, y,\, z,\, \psi,\, \theta,\, \phi)^T$, then the point belongs to the plane if:

$$B_{M_k P_k} \mathbf{x}_{M_k P_k} = z = 0$$

◇

2.3.4 Information Matrices

In the integration mechanism described above, matrix Q_n, the inverse of the covariance matrix P_n, accumulates information regarding the estimation of the location. For this reason, it is denominated *information matrix of the estimation.* The contribution of location information given by each pairing between an observation and a model feature is contained in matrix F_k, denominated *information matrix of the pairing.* These matrices are useful in determining how the inclusion of another pairing will improve the estimation of the location (chapters 4 and 5).

Algorithm 2.1 implements the integration process given here for a set of observed features. It is important to note that the integration process stated in this way is completely general. It allows to integrate geometric information about any type of geometric feature in a uniform way. The only information needed is the binding matrix of the involved feature or pairing.

```
FUNCTION estimate_object_location ($\mathcal{S}_h$)

; $\mathcal{S}_h = \{p_i = (e_i,\, m_i) : i = 1, \cdots, n\}$: initial pairings used to generate the hypothesis

; obtains an initial estimation for the location of the object
; returns the uncertain location vector of the object, $\mathbf{L}_{WO}$

    $\hat{\mathbf{x}}_{WO}$ := calculate_object_location ($p_1$, $p_2$);
    $\hat{\mathbf{d}}_O := 0$;
    $C_O := \infty$;
    $\mathbf{L}_{WO} := (\hat{\mathbf{x}}_{WO},\, \hat{\mathbf{d}}_O,\, C_O)$;
    FOR $p$ in $\mathcal{S}_h$ DO
        $\mathbf{L}_{WO}$ := refine_object_location ($\mathbf{L}_{WO}$, $p$);
    OD;

RETURN $\mathbf{L}_{WO}$;
END;


FUNCTION refine_object_location ($\mathbf{L}_{WO}$, $p$)

; $\mathbf{L}_{WO} = (\hat{\mathbf{x}}_{WO}, \hat{\mathbf{p}}_O, C_O)$: current estimated object location
; $p = (e,\, m)$: pairing to include in the hypothesis
; $e = (\mathbf{L}_{WE}, B_E),\ m = (\mathbf{x}_{OM}, B_M)$

; refines the object location with the given pairing, using the Extended information filter
; returns the uncertain location vector of the object, $\mathbf{L}_{WO}$

    $\hat{\mathbf{x}}_{EM} := \ominus\hat{\mathbf{x}}_{WE} \oplus \hat{\mathbf{x}}_{WO} \oplus \mathbf{x}_{OM}$;
    $\mathbf{h}_k := B_E\, \hat{\mathbf{x}}_{EM}$;
    $H_k := B_E J_{2\oplus}\{\hat{\mathbf{x}}_{EM}, 0\} B_E^T B_E J_{MO}$;
    $G_k := -B_E J_{1\oplus}\{0, \hat{\mathbf{x}}_{EM}\} B_E^T$;
    $S_k := C_E$;
    $F_k := H_k^T\, (G_k\, S_k\, G_k^T)^{-1}\, H_k$;
    $N_k := H_k^T\, (G_k\, S_k\, G_k^T)^{-1}\, h_k$;
    $Q := C_O^{-1} + F_k$;
    $M := Q\, \hat{\mathbf{d}}_O - N_k$;
    $\hat{\mathbf{d}}_O := Q^{-1}\, M$;
    $C_O := Q^{-1}$;

RETURN $\mathbf{L}_{WO}$;
END;
```

Algorithm 2.1: *Integrating Geometric Information*

2.4 Observation Modeling

In order to exemplify the use of the SPmodel and the integration mechanism, we will model several types of observations obtained with two types of sensors: mobile proximity and mobile 2D vision.

2.4.1 Modeling the Sensor Location

Due to the fact that both sensors are mounted on the robot gripper, the uncertainty of the observations given by them depends both on the *uncertainty of sensor location* and the *uncertainty of sensor measurements.* In this paragraph we model the uncertainty in the location of a sensor mounted on the robot gripper, due to robot positioning errors. Let G be a reference representing the location of the robot gripper in space. For simplicity, we will suppose that the robot location errors are independent on each axis. Thus, the uncertain location of the gripper with respect to a world reference, $\mathbf{L}_{WG}$, will be given by:

$$
\begin{aligned}
\hat{\mathbf{x}}_{WG} &= (x,\ y,\ z,\ \psi,\ \theta,\ \phi)^T \\
\mathbf{d}_G &= (d_x,\ d_y,\ d_z,\ d_\psi,\ d_\theta,\ d_\phi)^T \\
C_G &= \text{diag}(\sigma^2_{Rx},\ \sigma^2_{Ry},\ \sigma^2_{Rz},\ \sigma^2_{R\psi},\ \sigma^2_{R\theta},\ \sigma^2_{R\phi})
\end{aligned}
$$

In this case, the gripper does not have symmetries of continuous motion, so then $B_G = I_6$ and $\mathbf{d}_G = \mathbf{p}_G$. Let S be a reference associated to the location of the sensor. Let $\mathbf{x}_{GS}$ denote the relative transformation between the gripper and the sensor. According to (2.3), the uncertain location of the sensor with respect to the world reference is given by:

$$
\mathbf{L}_{WS} = \mathbf{L}_{WG} \oplus \mathbf{x}_{GS}
$$

where:

$$
\begin{aligned}
\hat{\mathbf{x}}_{WS} &= \hat{\mathbf{x}}_{WG} \oplus \mathbf{x}_{GS} \\
\mathbf{d}_S &= J_{SG}\,\mathbf{d}_G \\
C_S &= J_{SG}\ C_G\ J^T_{SG}
\end{aligned}
$$

Let E be a reference associated to a sensorial observation e, whose uncertain location with respect to the sensor reference is given by $\mathbf{L}_{SE} =$

$(\hat{\mathbf{x}}_{SE}, \hat{\mathbf{p}}_E^S, C_E^S)$. According to (2.4), the location of E with respect to W can be calculated as:

$$\mathbf{L}_{WE} = \mathbf{L}_{WS} \oplus \mathbf{L}_{SE}$$

where:

$$\begin{aligned}
\hat{\mathbf{x}}_{WE} &= \hat{\mathbf{x}}_{WG} \oplus \mathbf{x}_{GS} \oplus \hat{\mathbf{x}}_{SE} \\
\mathbf{p}_E &= B_E\, J_{ES}\, J_{SG}\, \mathbf{d}_G + \mathbf{p}_E^S \\
&= B_E\, J_{EG}\, \mathbf{d}_G + \mathbf{p}_E^S \\
C_E &= B_E\, J_{ES}\, J_{SG}\, C_G\, J_{SG}^T\, J_{ES}^T\, B_E^T + C_E^S \\
&= B_E J_{EG} C_G J_{EG}^T B_E^T + C_E^S \qquad (2.14)
\end{aligned}$$

Considering the estimations centered, we have $\hat{\mathbf{d}}_G = 0$ and $\hat{\mathbf{p}}_E^S = 0$, and thus $\hat{\mathbf{p}}_E = 0$. In this work we will consider robot positioning errors of 0.1 mm and orientation errors of 0.1 deg: $\sigma_{Rx}^2 = \sigma_{Ry}^2 = \sigma_{Rz}^2 = 0.01$ and $\sigma_{R\psi}^2 = \sigma_{R\theta}^2 = \sigma_{R\phi}^2 = 3.046174198 * 10^{-6}$.

2.4.2 Mobile Proximity

Consider the use of a proximity sensor mounted on the robot hand, as shown in figure 2.14. Let S be the reference associated to the location of the sensor, so that its z axis is aligned with the sensing direction. Let P be a reference associated to the location of the observed point. Its measurement error, expressed in reference S, can be expressed as a translation along the z axis of the sensor reference, and its angular error can be expressed as rotations around the x and y axes:

$$C_S = \mathrm{diag}(0, 0, \sigma_{Sz}^2, \sigma_{S\psi}^2, \sigma_{S\theta}^2, 0)$$

If the sensor is positioned at a distance d from the observed surface, then we have:

$$\hat{\mathbf{x}}_{SP} = (0,\ 0,\ d,\ -\pi/2,\ 0,\ 0)^T$$

In this case the subfeature is a point, so:

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

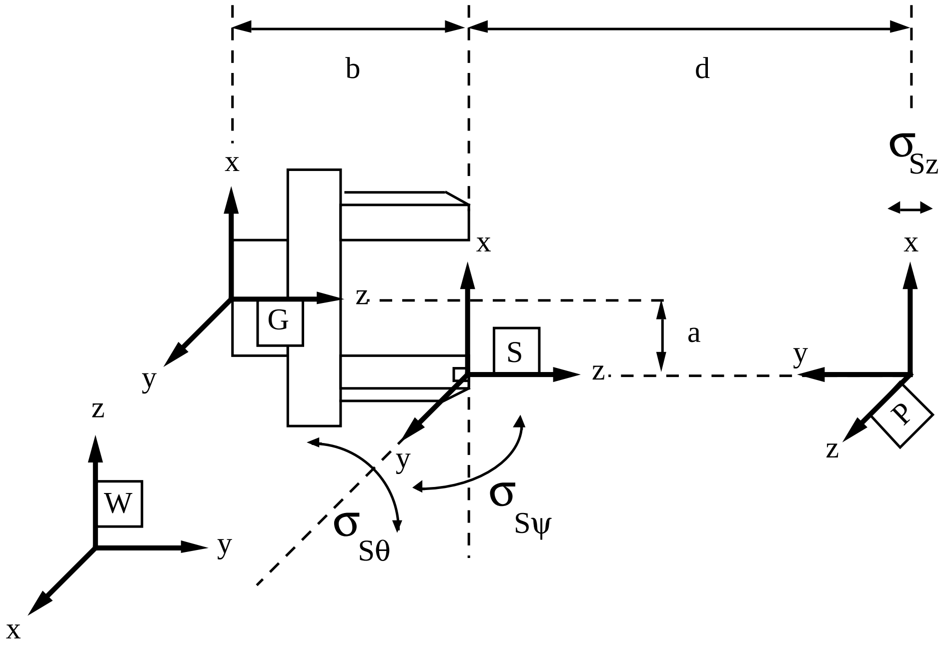


Figure 2.14: Proximity sensor

The Jacobian corresponding to the relative location between P and S is:

$$J_{PS} = \begin{bmatrix} 1 & 0 & 0 & 0 & d & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -d & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Thus, from (2.1), the uncertain location of the observed point with respect to the sensor reference is given by $\mathbf{L}_{SP} = (\hat{\mathbf{x}}_{SP}, \hat{\mathbf{p}}_P^S, C_P^S)$, where:

$$\begin{aligned} \hat{\mathbf{x}}_{SP} &= (0,\ 0,\ d,\ -\pi/2,\ 0,\ 0)^T \\ \hat{\mathbf{d}}_P^S &= J_{PS}\mathbf{d}_S \\ \hat{\mathbf{p}}_P^S &= B_P\mathbf{d}_P^S \\ &= (\hat{d}_x,\ \hat{d}_y,\ \hat{d}_z)^T \ = \ (0,\ 0,\ 0)^T \\ C_P^S &= B_P\, J_{PS}\, C_S J_{PS}^T\, B_P^T \end{aligned}$$

$$= \operatorname{diag}(d^2\sigma^2_{S\theta},\ \sigma^2_{Sz},\ d^2\sigma^2_{S\psi})$$

The relative location of the sensor with respect to the gripper is given by $\mathbf{x}_{GS} = (-a,\ 0,\ b,\ 0,\ 0,\ 0)^T$. Thus, according to (2.14), the uncertain location of the point with respect to the world reference is given by $\mathbf{L}_{WP} = (\hat{\mathbf{x}}_{WP}, \hat{\mathbf{p}}_P, C_P)$, where:

$$\begin{aligned}
\hat{\mathbf{x}}_{WP} &= \hat{\mathbf{x}}_{WG} \oplus \mathbf{x}_{GS} \oplus \hat{\mathbf{x}}_{SP} \\
&= (x + (b+d)\,(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) - a\ \cos\phi\cos\theta, \\
&\quad\ y + (b+d)\,(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) - a\ \sin\phi\cos\theta, \\
&\quad\ z + (b+d)\ \cos\theta\cos\psi + a\ \sin\theta, -\tan^{-1}(\cot\psi), \theta, \phi)^T \\
\hat{\mathbf{p}}_P &= 0 \\
C_P &= B_P\, J_{PG}\, C_G\, J^T_{PG}\, B^T_P + C^S_P \\
&= \begin{bmatrix} \sigma^2_x & \sigma_{xy} & 0 \\ \sigma_{xy} & \sigma^2_y & 0 \\ 0 & 0 & \sigma^2_z \end{bmatrix}
\end{aligned} \tag{2.15}$$

where:

$$\begin{aligned}
\sigma^2_x &= \sigma^2_{Rx} + (b+d)^2\,\sigma^2_{R\theta} + d^2\,\sigma^2_{S\theta} \\
\sigma^2_y &= \sigma^2_{Rz} + \sigma^2_{Sz} + a^2\sigma^2_{R\theta} \\
\sigma^2_z &= \sigma^2_{Ry} + (b+d)^2\,\sigma^2_{R\psi} + d^2\,\sigma^2_{S\psi} + a^2\sigma^2_{R\phi} \\
\sigma_{xy} &= -a\,(b+d)\,\sigma^2_{R\theta}
\end{aligned}$$

The value of the parameters corresponding to our sensors are: $a = 40$ mm, $b = 200$ mm, $d = 100$ mm, $\sigma_{S\psi} = \sigma_{S\theta} = 1.21$ deg, $\sigma_{Sz} = 1$ mm.

A proximity sensor can be used to observe different types of geometric features. In the following paragraphs we will derive observation models for two types of features: edges and planar surfaces.

Observing Edges

In observing an edge of length l, the sensor is positioned in a direction perpendicular to one of the adjacent planar surfaces [Montano 91], at a distance d, and n points are measured at a distance s from each other (figure 2.15). The location of the edge can be estimated by integrating the observed points using the information filter, as explained in section 2.3.1. In this case we have:

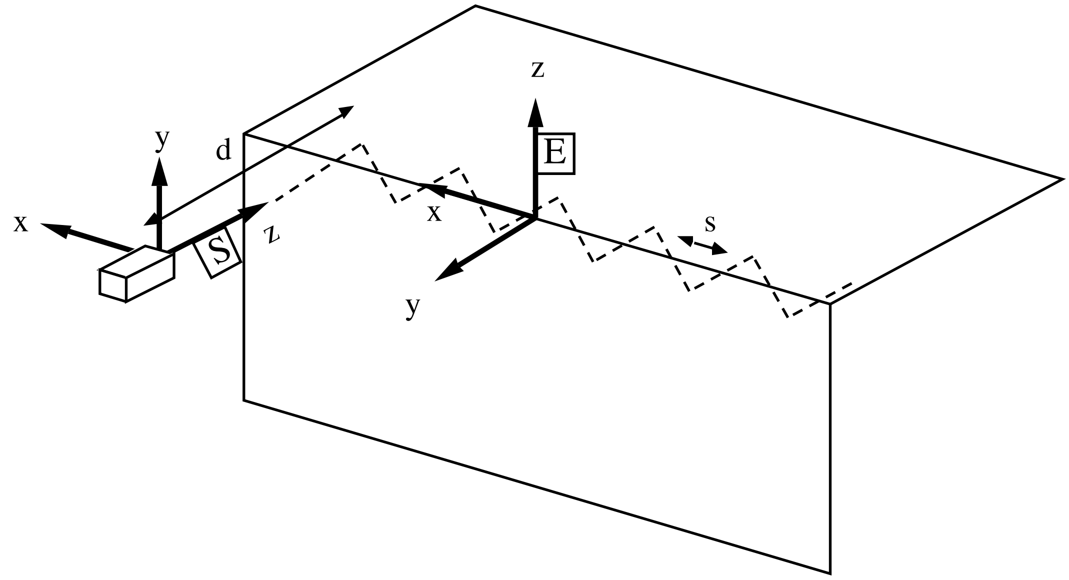


Figure 2.15: Observing an edge using proximity

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

All points will be observed from the same direction (observation is done using a sweeping movement), and we will place the associated reference in the center of the edge, giving it the orientation solidary with the sensing direction. Thus, we have that for each observation k (fig. 2.16):

$$\hat{\mathbf{x}}_{EP_k} = (r_k,\ e_{y_k},\ e_{z_k},\ 0,\ 0,\ 0)^T$$
$$S_k = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & 0 \\ \sigma_{xy} & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}$$

We suppose that each observation has the same orientation as the resulting edge because we consider $\sigma_{S\theta} = \sigma_{S\psi}$. The values of σ_x^2, σ_{xy}, σ_y^2 and σ_z^2 are those of (2.15). Given that an edge has a symmetry of translation along

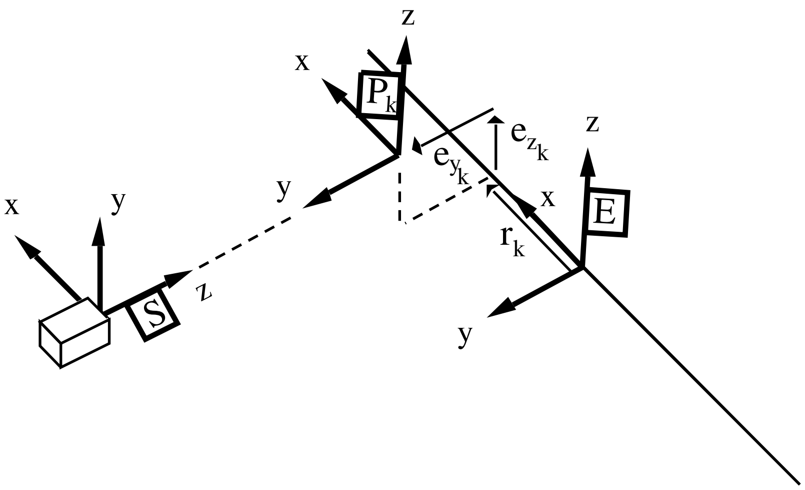


Figure 2.16: Relative location of the observed point P_k with respect to the estimated edge E

the x axis of its associated reference, an observed point on the edge cannot contribute information in that direction. It can only contribute information along the y and z axes. Thus, the pairing relationship can be described by the direct constraint using the following binding matrix:

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

In this way, from (2.9) we have:

$$\mathbf{h}_k = (e_{y_k}, e_{z_k})^T \; ; \; H_k = \begin{bmatrix} -1 & 0 & 0 & -r_k \\ 0 & -1 & r_k & 0 \end{bmatrix} \; ; \; G_k = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

From (2.8) we have that for each measure k to be integrated:

$$F_k = \begin{bmatrix} \frac{1}{\sigma_y^2} & 0 & 0 & \frac{r_k}{\sigma_y^2} \\ 0 & \frac{1}{\sigma_z^2} & -\frac{r_k}{\sigma_z^2} & 0 \\ 0 & -\frac{r_k}{\sigma_z^2} & \frac{r_k^2}{\sigma_z^2} & 0 \\ \frac{r_k}{\sigma_y^2} & 0 & 0 & \frac{r_k^2}{\sigma_y^2} \end{bmatrix} \; ; \; N_k = \left(-\frac{e_{y_k}}{\sigma_y^2}, -\frac{e_{z_k}}{\sigma_z^2}, \frac{r_k e_{z_k}}{\sigma_z^2}, -\frac{r_k e_{y_k}}{\sigma_y^2} \right)^T$$

Thus, from (2.7), when all n measures have been integrated we have:

$$
\begin{aligned}
Q_n &= \begin{bmatrix} \frac{n}{\sigma_y^2} & 0 & 0 & \frac{\sum_{k=1}^n r_k}{\sigma_y^2} \\ 0 & \frac{n}{\sigma_z^2} & -\frac{\sum_{k=1}^n r_k}{\sigma_z^2} & 0 \\ 0 & -\frac{\sum_{k=1}^n r_k}{\sigma_z^2} & \frac{\sum_{k=1}^n r_k^2}{\sigma_z^2} & 0 \\ \frac{\sum_{k=1}^n r_k}{\sigma_y^2} & 0 & 0 & \frac{\sum_{k=1}^n r_k^2}{\sigma_y^2} \end{bmatrix} \\
M_n &= \left(-\frac{\sum_{k=1}^n e_{y_k}}{\sigma_y^2}, -\frac{\sum_{k=1}^n e_{z_k}}{\sigma_z^2}, \frac{\sum_{k=1}^n r_k e_{z_k}}{\sigma_z^2}, -\frac{\sum_{k=1}^n r_k e_{y_k}}{\sigma_y^2} \right)^T
\end{aligned}
$$

Since we will locate the reference frame associated to the edge in its center, we have $\sum_{k=1}^n r_k = 0$, $\sum_{k=1}^n r_k^2 \simeq \frac{n^3\, s^2}{12}$. Therefore, Q_n becomes:

$$
Q_n = \begin{bmatrix} \frac{n}{\sigma_y^2} & 0 & 0 & 0 \\ 0 & \frac{n}{\sigma_z^2} & 0 & 0 \\ 0 & 0 & \frac{n^3\, s^2}{12\sigma_z^2} & 0 \\ 0 & 0 & 0 & \frac{n^3\, s^2}{12\sigma_y^2} \end{bmatrix}
$$

Thus, the final result of the integration process yields:

$$
\begin{aligned}
\hat{\mathbf{p}}_E &= \hat{\mathbf{x}}_n = \left(-\frac{\sum_{k=1}^n e_{y_k}}{n}, -\frac{\sum_{k=1}^n e_{z_k}}{n}, \frac{12\sum_{k=1}^n r_k e_{z_k}}{n^3\, d^2}, -\frac{12\sum_{k=1}^n r_k e_{y_k}}{n^3\, d^2} \right)^T \\
C_E &= P_n = \frac{1}{n} \begin{bmatrix} \sigma_y^2 & 0 & 0 & 0 \\ 0 & \sigma_z^2 & 0 & 0 \\ 0 & 0 & \frac{12}{n^2\, s^2}\,\sigma_z^2 & 0 \\ 0 & 0 & 0 & \frac{12}{n^2\, s^2}\,\sigma_y^2 \end{bmatrix} \qquad (2.16)
\end{aligned}
$$

It is interesting to note that the covariance of the estimation depends on n and s. This means that if we observe more points—n grows—or if we obtain more distant points—s grows—we obtain a more precise estimation of the edge location. Therefore, the precision of the observation depends on the sensor precision (represented by σ_y and σ_z), and by the adopted sensing strategy (represented by n and s). Also, taking into account that if we observe the whole edge, we have that $n\, s = l$, and thus the precision of the observation also depends on the edge dimensions. Longer edges will be more precise because n will be greater.

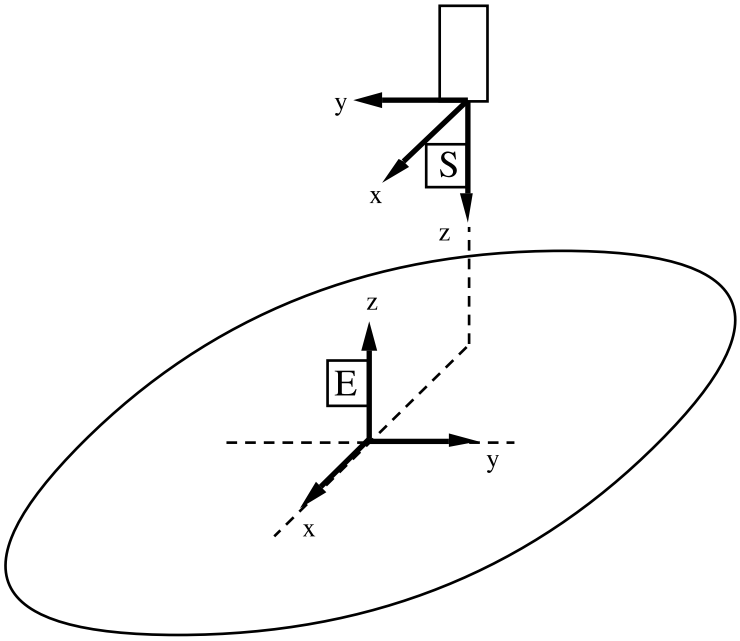


Figure 2.17: Observing a plane using proximity

Observing Planar Surfaces

Several strategies can be used to observe a planar surface with a proximity sensor [Montano 91]. The one we model in this section consists in positioning the sensor in a direction near to the normal to the surface and performing a cross-shaped movement around the center (figure 2.17). For each movement, n points are measured at a distance s from each other. Again, we can estimate the location of the plane by integrating the observed points using the information filter, as explained in section 2.3.1. A plane has symmetries along its x and y axes, and around its z axis ($T_{xy}\, R_z$). Thus, its self-binding matrix is:

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

Given that all points will be observed from the same direction, and placing

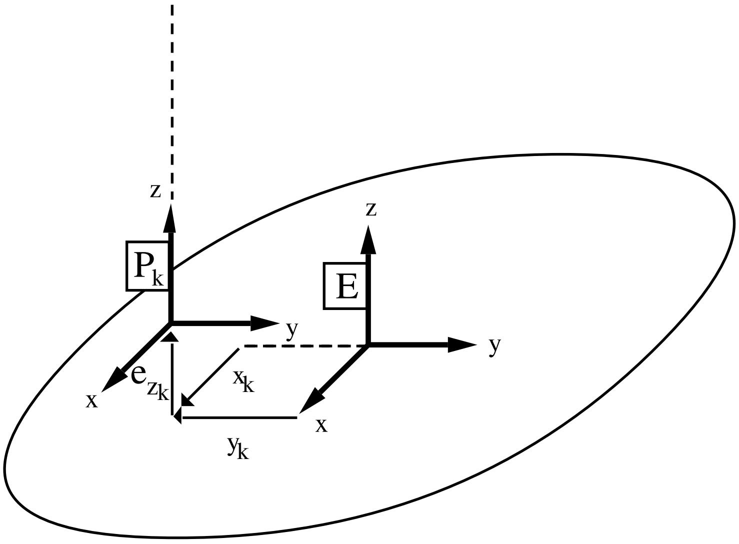


Figure 2.18: Relative location of the observed point P_k with respect to the estimated plane E

the associated reference of the plane in the center of the cross, we have that for each observation k (fig. 2.18):

$$\hat{\mathbf{x}}_{EP_k} = (x_k,\ y_k,\ e_{z_k},\ 0,\ 0,\ 0)^T$$

$$S_k = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & 0 \\ \sigma_{xy} & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}$$

An observed point on the plane can only contribute information on the z direction of the plane reference. Thus, the pairing relationship can be described by the direct constraint using the following binding matrix:

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

In this way, from (2.9) we have:

$$\mathbf{h}_k = e_{z_k} \ ; \ H_k = \begin{bmatrix} 1 & y_k & -x_k \end{bmatrix} \ ; \ G_k = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

From (2.8) we have that for each measure k to be integrated:

$$F_k = \frac{1}{\sigma_z^2}\begin{bmatrix} 1 & y_k & -x_k \\ y_k & y_k^2 & -y_k x_k \\ -x_k & -y_k x_k & x_k^2 \end{bmatrix} \; ; \; N_k = \frac{1}{\sigma_z^2}\,(e_{z_k}, y_k\, e_{z_k}, -x_k\, e_{z_k})^T$$

Thus, from (2.7), when all n measures have been integrated we have:

$$\begin{aligned} Q_n &= \frac{1}{\sigma_z^2}\begin{bmatrix} n & \sum_{k=1}^n y_k & -\sum_{k=1}^n x_k \\ \sum_{k=1}^n y_k & \sum_{k=1}^n y_k^2 & -\sum_{k=1}^n x_k\, y_k \\ -\sum_{k=1}^n x_k & -\sum_{k=1}^n x_k\, y_k & \sum_{k=1}^n x_k^2 \end{bmatrix} \\ M_n &= \frac{1}{\sigma_z^2}\left(-\sum_{k=1}^n e_{z_k}, -\sum_{k=1}^n y_k\, e_{z_k}, \sum_{k=1}^n x_k\, e_{z_k}\right)^T \end{aligned}$$

Since we will locate the reference frame associated to the plane in the center of the cross, we have $\sum_{k=1}^n x_k = \sum_{k=1}^n y_k = \sum_{k=1}^n x_k\, y_k = 0$, and $\sum_{k=1}^n x_k^2 = \sum_{k=1}^n y_k^2 \simeq \frac{n^3\, s^2}{96}$. Therefore, Q_n becomes:

$$Q_n = \frac{1}{\sigma_z^2}\begin{bmatrix} n & 0 & 0 \\ 0 & \frac{n^3\, s^2}{96} & 0 \\ 0 & 0 & \frac{n^3\, s^2}{96} \end{bmatrix}$$

The result of the integration process is:

$$\begin{aligned} \hat{\mathbf{p}}_E &= \hat{\mathbf{x}}_n = \left(-\frac{\sum_{k=1}^n e_{z_k}}{n}, -\frac{96\sum_{k=1}^n y_k e_{z_k}}{n^3\, s^2}, \frac{96\sum_{k=1}^n x_k e_{z_k}}{n^3\, s^2}\right)^T \\ C_E &= P_n = \sigma_z^2 \begin{bmatrix} \frac{1}{n} & 0 & 0 & \\ 0 & \frac{96}{n^3\, s^2} & 0 & \\ 0 & 0 & 0 & \frac{96}{n^3\, s^2} \end{bmatrix} \end{aligned} \tag{2.17}$$

2.4.3 Mobile Vision

Consider the use of a camera mounted on the robot hand, as shown in figure 2.19. Concentrating on the observation of straight edges, if we only take one image, we obtain a *subfeature* associated to the edge that we call a *2D edge*. We can also use two images to completely determine the location of the edge, obtaining a *3D edge*. The models for these two types of observations are given below.

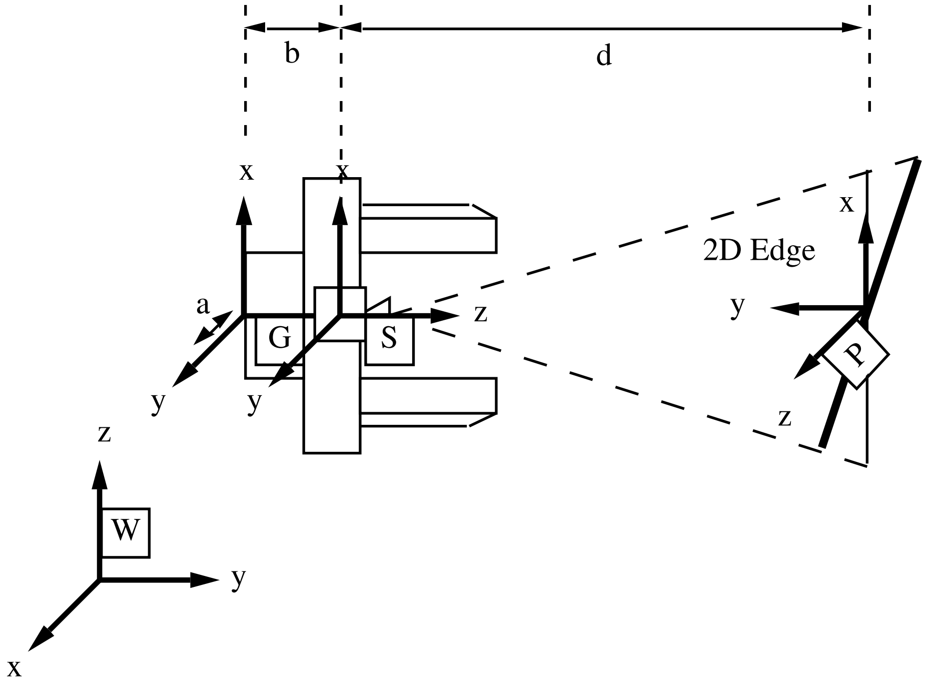


Figure 2.19: Camera-in-hand vision system

Observing 2D Edges

Consider the observation of and edge located at a distance d from the camera (figure 2.19). Taking only one image, the information we obtain about the edge corresponds to a plane where it rests, called a *2D edge.* We associate to it a reference whose z axis is perpendicular to the plane formed by the edge and the optical center of the camera. This type of feature has the same symmetries of a plane ($T_{xy}\, R_z$). Thus, in this case we have:

$$\begin{aligned}
\hat{\mathbf{x}}_{SP} &= (0,\ 0,\ d,\ -\pi/2,\ 0,\ 0)^T \\
\mathbf{p}_P &= (d_z,\ d_\psi,\ d_\theta)^T \\
B_P &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}
\end{aligned}$$

The values of d_z and d_θ represent the position and orientation error of the edge in the image plane. Since the 2D edge is solidary to the camera position, the value of d_ψ can be calculated from d_z as:

$$d_\psi = -\frac{d_z}{d}$$

Thus, we will use the information filter to estimate only the values of d_z and d_θ. For this reason, we will take the binding matrix of the feature to be estimated as:

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

Each observed pixel only contributes information in the direction of the z axis of the 2D edge (fig. 2.20). Thus, we will take:

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

Suppose the location error of each pixel in the z direction of P is Gaussian, with expected value 0 and standard deviation σ_p in mm (this value depends on the scale factor s (mm /pixel) of the image plane at distance d). We have that for each observation k (fig. 2.20):

$$\hat{\mathbf{x}}_{EP_k} = \left(r_k,\ 0,\ e_{z_k},\ 0,\ 0,\ \pi/2 + \tan^{-1}(r_k/d) \right)^T$$

Since the pixel can only contribute information on the z direction of the plane reference, the pairing relationship can be described using the following binding matrix:

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

From (2.9) we have:

$$\mathbf{h}_k = e_{z_k} \ ;\ H_k = \begin{bmatrix} 1 & -r_k \end{bmatrix} \ ;\ G_k = 1$$

From (2.8) we have that for each measure k to be integrated:

$$F_k = \frac{1}{\sigma_z^2} \begin{bmatrix} 1 & -r_k \\ -r_k & r_k^2 \end{bmatrix} \ ;\ N_k = \frac{1}{\sigma_z^2} \left(e_{z_k}, r_k\, e_{z_k} \right)^T$$

Thus, from (2.7), when all n measures have been integrated we have:

$$Q_n = \frac{1}{\sigma_z^2} \begin{bmatrix} n & -\sum_{k=1}^n r_k \\ -\sum_{k=1}^n r_k & \sum_{k=1}^n r_k^2 \end{bmatrix} \ ;\ M_n = \frac{1}{\sigma_z^2} \left(-\sum_{k=1}^n e_{z_k}, -\sum_{k=1}^n r_k\, e_{z_k} \right)^T$$

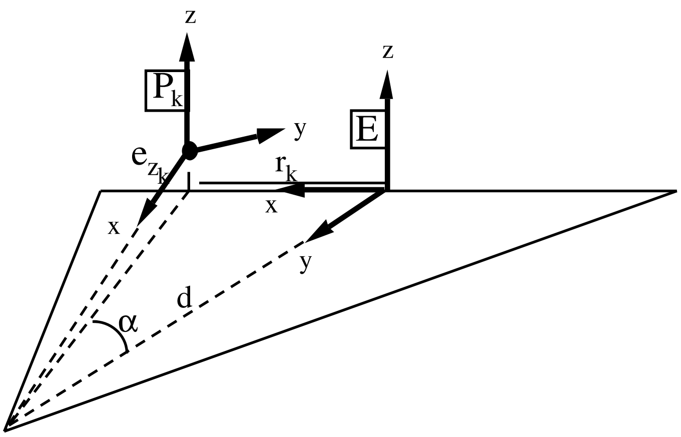


Figure 2.20: Relative location of the observed pixel P_k with respect to the estimated 2D edge E

Since we will locate the reference frame associated to the 2D edge in the midpoint of the observed pixels, we have $\sum_{k=1}^{n} r_k = 0$, and $\sum_{k=1}^{n} r_k^2 \simeq \frac{n^3 s^2}{12}$. Therefore, Q_n becomes:

$$Q_n = \frac{1}{\sigma_z^2}\begin{bmatrix} n & 0 \\ 0 & \frac{n^3 s^2}{12} \end{bmatrix}$$

The result of the integration process is:

$$\hat{\mathbf{x}}_n = \left(-\frac{\sum_{k=1}^{n} e_{z_k}}{n}, -\frac{12\sum_{k=1}^{n} r_k e_{z_k}}{n^3 s^2}\right)^T \quad ; \quad P_n = \begin{bmatrix} \frac{\sigma_z^2}{n} & 0 \\ 0 & \frac{12\sigma_z^2}{n^3 s^2} \end{bmatrix}$$

Including the component of the perturbation vector corresponding to ψ, the estimated location of the plane e will be given by $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E, C_E)$, where:

$$C_E = \frac{\sigma_z^2}{n}\begin{bmatrix} 1 & -\frac{1}{d} & 0 \\ -\frac{1}{d} & \frac{1}{d^2} & \\ 0 & 0 & \frac{12}{n^2 s^2} \end{bmatrix}$$

Given that the relative location between the gripper and the sensor is $\mathbf{x}_{GS} = \text{Trans}(0, a, b)$, and taking into account the uncertainty in the location of the gripper, according to (2.14):

$$C_E = \begin{bmatrix} \sigma_z^2 & \sigma_{z\psi} & 0 \\ \sigma_{z\psi} & \sigma_\psi^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \tag{2.18}$$

where:

$$\begin{aligned}
\sigma_z^2 &= \sigma_{Ry}^2 + (b+d)^2\,\sigma_{R\psi}^2 + \frac{\sigma_{Sp}^2}{n} \\
\sigma_{z\psi} &= -(b+d)\,\sigma_{R\psi}^2 - \frac{\sigma_{Sp}^2}{nd} \\
\sigma_\psi^2 &= \sigma_{R\psi}^2 + \frac{\sigma_{Sp}^2}{nd^2} \\
\sigma_\theta^2 &= \sigma_{R\phi}^2 + \frac{12\,\sigma_{Sp}^2}{n^3\,s^2}
\end{aligned}$$

Observing 3D Edges

We can use the camera to estimate the location of an edge by obtaining two 2D edges from two directions such that their relative angle is λ. We use the two observations to analytically determine the location of the edge, which lies at the intersection of the two projecting planes (fig. 2.21). Thus, we can associate a reference E, solidary with the first observation, and locate the references associated to the two observations P_1 and P_2 so that:

$$\hat{\mathbf{x}}_{P_1E} = (0,\ 0,\ 0,\ 0,\ 0,\ 0)^T$$

$$\hat{\mathbf{x}}_{P_2E} = (0,\ 0,\ 0,\ \lambda,\ 0,\ 0)^T$$

To estimate the location of the edge we use the information filter taking the inverse constraint as measurement function. Each observed 2D edge contributes information to the edge in the direction of the z axis and around the y axis (fig. 2.22). Therefore, the pairing relationship can be described by the following binding matrix:

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

In this way, from (2.9) we have:

$$\mathbf{h}_k = (0,0)^T$$

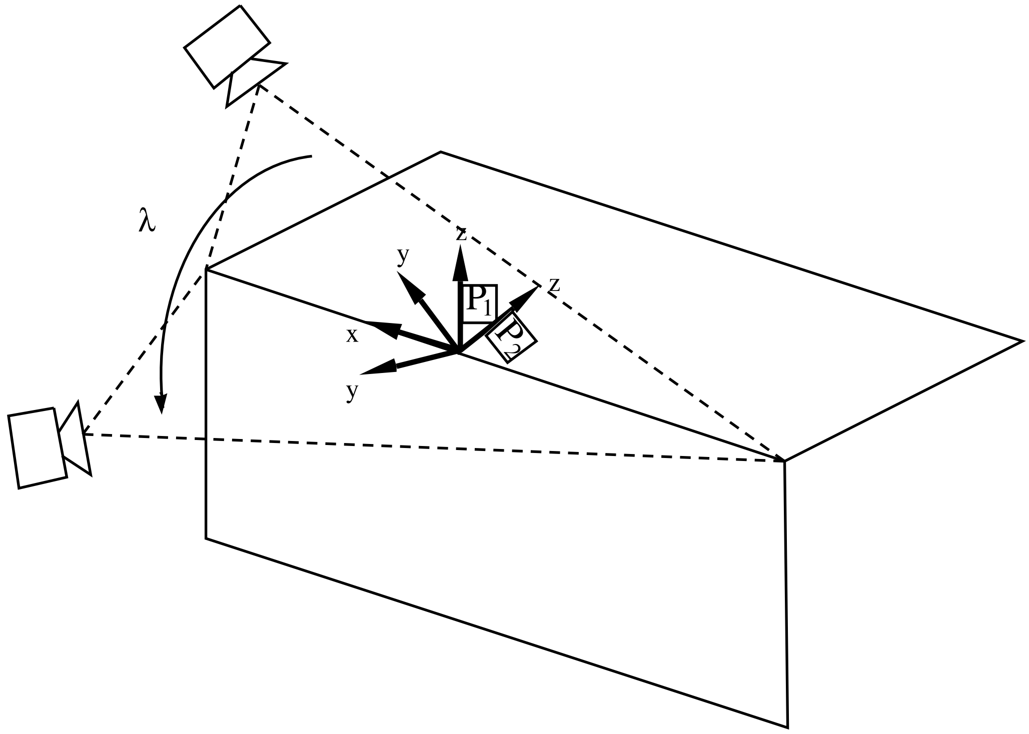


Figure 2.21: Observing a 3D edge

$$H_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$H_2 = \begin{bmatrix} \sin\lambda & \cos\lambda & 0 & 0 \\ 0 & 0 & \cos\lambda & -\sin\lambda \end{bmatrix}$$

$$G_1 = G_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

From (2.8) we have that the information matrices of the pairings are:

$$F_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_z^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_\theta^2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

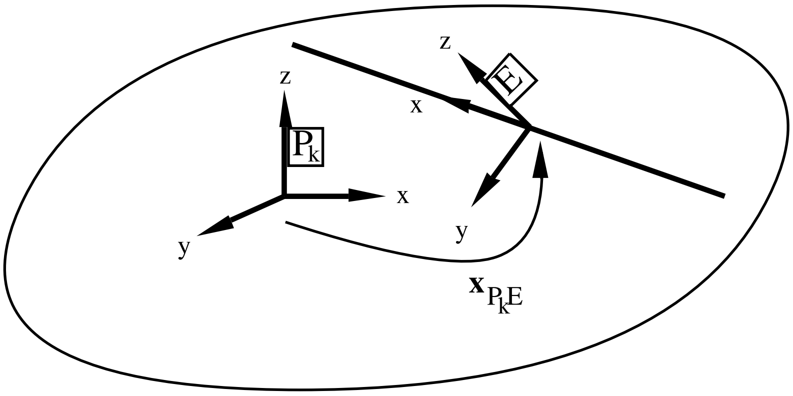


Figure 2.22: The relative location between a plane and an edge contained in the plane is given by $\mathbf{x}_{EP_k} = (x,\, y,\, 0,\, \psi,\, 0,\, \phi)^T$

$$F_2 = \begin{bmatrix} \frac{\sin^2\lambda}{\sigma_z^2} & \frac{\sin\lambda\cos\lambda}{\sigma_z^2} & 0 & 0 \\ \frac{\sin\lambda\cos\lambda}{\sigma_z^2} & \frac{\cos^2\lambda}{\sigma_z^2} & 0 & 0 \\ 0 & 0 & \frac{\cos^2\lambda}{\sigma_\theta^2} & \frac{-\sin\lambda\cos\lambda}{\sigma_\theta^2} \\ 0 & 0 & \frac{-\sin\lambda\cos\lambda}{\sigma_\theta^2} & \frac{\sin^2\lambda}{\sigma_\theta^2} \end{bmatrix}$$

Thus, the information matrix of the estimation is:

$$Q_2 = \begin{bmatrix} \frac{\sin^2\lambda}{\sigma_z^2} & \frac{\sin\lambda\cos\lambda}{\sigma_z^2} & 0 & 0 \\ \frac{\sin\lambda\cos\lambda}{\sigma_z^2} & \frac{1+\cos^2\lambda}{\sigma_z^2} & 0 & 0 \\ 0 & 0 & \frac{1+\cos^2\lambda}{\sigma_\theta^2} & -\frac{\sin\lambda\cos\lambda}{\sigma_\theta^2} \\ 0 & 0 & -\frac{\sin\lambda\cos\lambda}{\sigma_\theta^2} & \frac{\sin^2\lambda}{\sigma_\theta^2} \end{bmatrix}$$

Thus, the resulting covariance matrix of the estimated edge is:

$$C_E = \begin{bmatrix} \frac{(1+\cos^2\lambda)\sigma_z^2}{\sin^2\lambda} & -\frac{\cos\lambda\sigma_z^2}{\sin\lambda} & 0 & 0 \\ -\frac{\cos\lambda\sigma_z^2}{\sin\lambda} & \sigma_z^2 & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 & \frac{\cos\lambda\sigma_\theta^2}{\sin\lambda} \\ 0 & 0 & \frac{\cos\lambda\sigma_\theta^2}{\sin\lambda} & \frac{\left(1+\cos^2\lambda\right)\sigma_\theta^2}{\sin^2\lambda} \end{bmatrix} \tag{2.19}$$

It is interesting to see how the values of the covariance matrix depend on the value of λ. When $\lambda = \pi/2$, the two projecting planes are orthogonal, and the resulting covariance matrix of the edge is:

$$C_E = \begin{bmatrix} \sigma_z^2 & 0 & 0 & 0 \\ 0 & \sigma_z^2 & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & 0 & \sigma_\theta^2 \end{bmatrix}$$

If the value of lambda is smaller, the values of the covariance matrix become larger, and thus the estimation becomes less precise. When $\lambda = 0$, the two projecting planes are parallel and the covariance matrix contains infinite terms (the edge location cannot be determined). The concept of *precision* of an estimation is studied in chapter 4. Its relationship with sensor location is studied in chapter 5.

The parameters of this sensor are $a = 70$ mm, $b = 140$ mm. For an edge located at a distance $d = 160$ mm, the sensor parameters are $s = 1$ mm, and $\sigma_{Sp} = 0.51$ mm.

2.5 Conclusions

In this chapter we have analyzed the advantages of using probabilistic models over set-based models in representing uncertain geometric information. We have described the SPmodel, a probabilistic model to represent uncertain geometric information that combines the use of probability theory to represent the imprecision associated to the location of a geometric entity, and the theory of symmetries to represent the partiality due to the characteristics of the geometric element. We have also described the use of the SPmodel in the definition of a general mechanism for the integration of uncertain geometric information, which is a nonrecursive formulation of the information filter. We have exemplified the use of the SPmodel and the integration mechanism by modeling observations given by mobile proximity and mobile 2D vision sensors. These observation models will help us illustrate the proposed perception and recognition strategies.

The SPmodel and integration mechanism constitute the backbone of this work. The concept of symmetries of a geometric element help us to determine the geometric relations between a pair of features, which constitute the fundamental concept related to constraint validation (chapter 3). The study of the information matrix of the estimation and of a pairing is used to define the concepts of *precision* of an object-location hypothesis and *relevance* of a pairing and of a hypothesis (chapter 4). The information matrix of a pairing also allow us to relate the precision of a given sensor, and its location relative to the perceived feature, with the amount of location information that the observation will *contribute* to the estimation of the object location. We make use of this concept of contribution to compare the potential benefit of using different sensors for a given perception task, and to determine the sensor location such that this contribution is maximal (chapter 5). These concepts are fundamental to study several aspects of the complexity of the recognition process, and how to reduce it (chapter 6).

Chapter 3

Geometric Constraints

Summary

Geometric constraints constitute one of the most important sources of information to establish the validity of an interpretation of a set of observations with respect to an object model. This validation mechanism allows to reduce the complexity of the recognition process by limiting the number of hypotheses that the system must verify. In this chapter we present a general procedure to compute and validate geometric constraints between uncertain geometric features [Neira 93a]. The proposed procedure allows to obtain the geometric relations between any pair of geometric elements in a systematic way, with an explicit consideration of the uncertainty due to the use of different sensors. This makes this constraint validation mechanism well suited for multisensor systems.

3.1 Introduction

The object recognition process based on the matching between model features and sensor observations is of exponential complexity. One of the fundamental ideas to reduce this complexity is the use of geometric constraints: the validation that geometric relations between model features are satisfied in the observations we are trying to match with them. We can classify geometric constraints into two categories:

- *Location independent constraints*, which can be validated without having an estimation of the location of the object. They include *unary* constraints: dimensions such as edge length, or surface area, and *binary* constraints: distances and angles between features.

- *Location dependent constraints*, based on the availability of the object location. The fundamental constraint of this type is *rigidity*: the location of an object in the scene determines the location of its features in the scene.

Geometric relations are a set of parameters that derive from the geometry of each feature, and from the relative location between features. In systems that rely on one sensor and only consider a specific type of feature (stereo vision and edges, for example), these parameters are easily derived. However, in multisensor systems, where different sensors or combinations of sensors can give geometric information of diverse type, such as dihedrals, corners or circles, we have the problem of deriving them for each pair of geometric elements. Thus, the first question that we try to answer here is: *what geometric relations can be established between each combination of geometric elements?*

The validation of geometric constraints is complicated by the diverse nature of sensorial information and its uncertainty. Given that we are dealing with imprecise sensorial information, the second important problem we must take into account is: *how can we estimate and validate geometric relations under uncertainty?*

Considerable work has been done on the use geometric constraints in object recognition. In [Grimson 90a], Grimson proposes a recognition process which relies on the *identifying before locating* approach, in which simple constraints are used to validate the consistency between a set of observations and an object model, pruning inconsistent interpretations. Grimson develops unary and binary constraints for features such as edges, circular arcs, or cylinders. These constraints are independent of the object location and thus can be calculated without estimating it. The Handey system [Lozano-Pérez 87] uses these types of geometric constraints, applied to edges, in its matching algorithm to locate objects. One of the main drawbacks of these approaches is that uncertainty is modeled using error bounds. As we have seen in chapter 2, this can result in high computational complexity, or it can lead to conservative estimations of errors.

Porrill [Porrill 88] uses a probabilistic model to establish a mechanism to combine information from multiple sensors that can be used to locate an object given its wireframe representation and a set of observations. Location independent geometric constraints, such as distances and angles between edges, are used to impose consistency. Since most geometric relations are nonlinear functions of the relative feature location, a first-order approximation is used. Sagüés and Montano [Sagüés 92b] propose a similar approach. They develop constraint validation mechanisms for vertices, edges and planar surfaces. In these works, the expressions for the geometric constraints between specific types of features are derived. However, no general constraint validation mechanism is proposed.

An additional drawback of the works of Grimson and Porrill is the fact that location independent constraints are less discriminant than location dependent ones, because they only validate *local consistency*, and thus many inconsistent interpretations survive up to object location estimation. In the work of Faugeras [Faugeras 86], the *identifying while locating* approach is proposed. Emphasis is made on the use of *rigidity* in the recognition process as early as possible, since it is the only constraint that guarantees global consistency. In general, these and other works [Bolles 86, Crowley 87] consider a limited number of different features, and intuitively derive a set of geometric constraints to be used in hypothesis validation. The fact that the location of each type of feature has a different representation makes it complex to derive a general method to measure and validate geometric constraints.

We propose a geometric validation mechanism that is based in the SPmodel [Neira 93a]. The probabilistic representation of uncertainty adopted in the SPmodel allows to reflect the different capabilities of sensors, and thus, it is adequate for multisensor systems. The explicit consideration of the symmetries of a geometric element allows us to develop a general method to *systematically* deduce the geometric relations between any combination of geometric features, not limiting the system to a restricted set of features or sensors. In our work, location independent and location dependent constraints are represented in a uniform way. This allows us to define a general mechanism for the validation of geometric constraints under uncertainty.

This chapter is organized as follows: in section 3.2 we show how to systematically deduce location independent geometric relations and express them using a uniform representation. This allows us to define a general mechanism for the validation of location independent geometric constraints under uncertainty. Section 3.3 is devoted to the study of location dependent geometric relations for all types of geometric features.

3.2 Location Independent Geometric Relations

In this section we study *binary* geometric relations: the geometric relations that determine the relative location of the two geometric features. In general, *unary* geometric relations are particular to each type of geometric feature, and easy to validate. Furthermore, we only consider parameters that are invariant under partial occlusion, to allow their application to cluttered scenes.

3.2.1 Deriving Binary Relations

In general, binary geometric relations are nonlinear functions of the relative location of the involved geometric features [Sagüés 92b]. Thus, in estimating the value of these relations and their uncertainty, a first order approximation is necessary. We propose an approach consisting in systematically obtaining a linear representation for binary relations, from which estimating their value and its uncertainty is straightforward. Consider the following example.

Example 3.1: *Geometric relations between two edges*
Consider a pair of model edges whose location is represented by references A and B respectively (fig. 3.1). In order to concentrate on occlusion-invariant parameters, we consider the edges of infinite length. Intuitively, we can see that their relative location is defined by two parameters: their perpendicular distance, and the angle between them. Thus, the geometric relations between the two edges are $\mathbf{r} = (d,\ \alpha)$. Given $\mathbf{x}_{AB} = (x,\ y,\ z,\ \psi,\ \theta,\ \phi)^T$, the relative location vector between the two edges, the distance d and the angle α can be calculated as follows:

$$\begin{aligned} d &= \frac{y \sin\theta + z \sin\phi \cos\theta}{\sqrt{1 - \cos^2\theta \cos^2\phi}} \\ \alpha &= \tan^{-1}\left(\frac{\sqrt{1 - \cos^2\theta \cos^2\phi}}{\cos\theta \cos\phi} \right) \end{aligned}$$

If we consider that A and B express the uncertain location of observed edges,

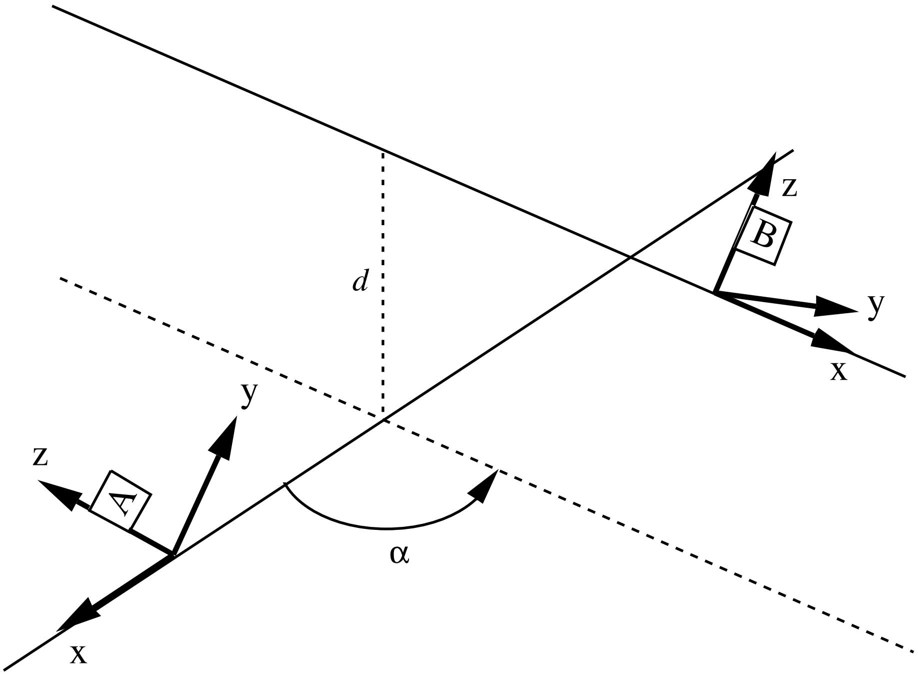


Figure 3.1: Geometric relations between two edges

in order to compute the uncertainty of d due to the uncertainty of $\mathbf{x}_{AB}$, we need to use a first order approximation of d:

$$\begin{aligned}\hat{d}(\mathbf{x}_{AB}) = d(\hat{\mathbf{x}}_{AB}) &+ \frac{\partial d}{\partial y}(\hat{\mathbf{x}}_{AB})\,(y-\hat{y}) \\ &+ \frac{\partial d}{\partial z}(\hat{\mathbf{x}}_{AB})\,(z-\hat{z}) + \frac{\partial d}{\partial \theta}(\hat{\mathbf{x}}_{AB})\,(\theta-\hat{\theta}) + \frac{\partial d}{\partial \phi}(\hat{\mathbf{x}}_{AB})\,(\phi-\hat{\phi})\end{aligned}$$

This approach has two fundamental drawbacks:

- The expressions of these approximations are particular for each case, and can be considerably complex.
- It is not always clear which geometric relations can be measured between a pair of features.

An alternative method is to find two *aligned* references $\bar{A}$ and $\bar{B}$ which equivalently describe the location of the edges, such that d and α are linear functions of $\mathbf{x}_{\bar{A}\bar{B}} = (0,\ 0,\ \bar{z},\ 0,\ 0,\ \bar{\phi})^T$ (fig. 3.2):

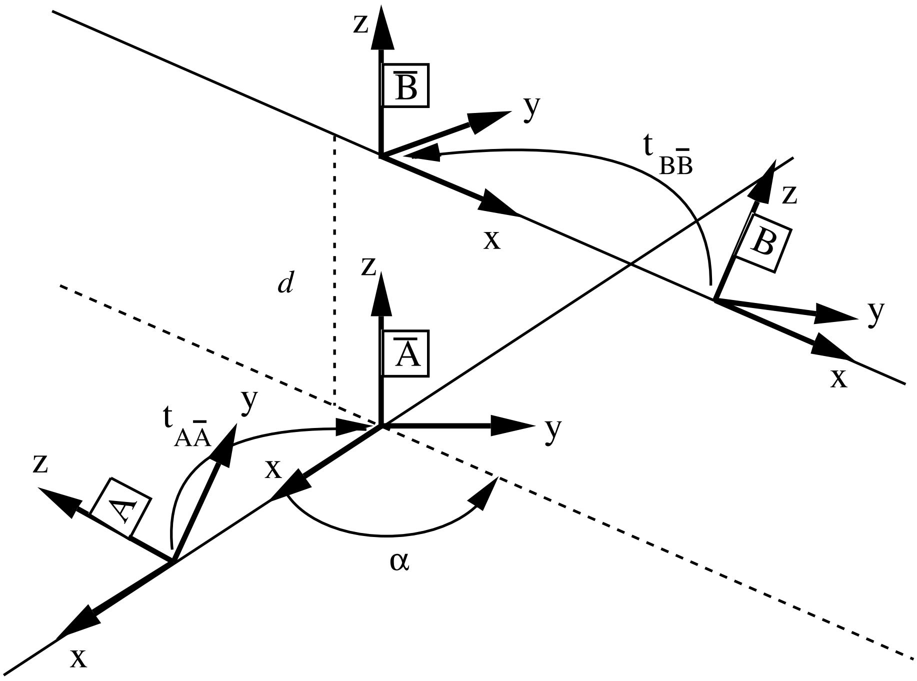


Figure 3.2: Reference alignment for two edges

$$\begin{aligned} d &= \bar{z} \\ \alpha &= \bar{\phi} \end{aligned}$$

Note that the aligning transformations $t_{A\bar{A}}$ and $t_{B\bar{B}}$ belong to the symmetries of their respective geometric elements: $T_x\, R_x$. ◇

Generalization: The idea of finding a pair of aligning transformations belonging to the symmetries of the corresponding geometric elements can be generalized in the following way: let t_{WA} and t_{WB} express the location of two geometric elements A and B with respect to a base reference W, and let t_{AB} be the transformation that expresses the relative location between both elements. Let $\mathcal{S}_A$ and $\mathcal{S}_B$ denote the subgroups of symmetries of the corresponding geometric elements. Then, the sets of transformations that equivalently describe the locations of A and B are:

$$\mathcal{L}_A = t_{WA} \cdot \mathcal{S}_A \quad ; \quad \mathcal{L}_B = t_{WB} \cdot \mathcal{S}_B$$

The set of all possible transformations between the references associated to the two geometric elements is given by:

$$\begin{aligned}
\mathcal{L}_{AB} &= \mathcal{L}_A^{-1} \cdot \mathcal{L}_B \\
&= (t_{WA} \cdot \mathcal{S}_A)^{-1} \cdot (t_{WB} \cdot \mathcal{S}_B) \\
&= S_A^{-1} \cdot t_{WA}^{-1} \cdot t_{WB} \cdot \mathcal{S}_B \\
&= \mathcal{S}_A \cdot t_{AW} \cdot t_{WB} \cdot \mathcal{S}_B \\
&= \mathcal{S}_A \cdot t_{AB} \cdot \mathcal{S}_B
\end{aligned}$$

Thus, the set $\mathcal{L}_{AB}$ can be characterized in the following way:

$$\mathcal{L}_{AB} = \left\{ t_{\bar{A}\bar{B}} = t_{A\bar{A}}^{-1} \cdot t_{AB} \cdot t_{B\bar{B}} \;\mid\; t_{A\bar{A}} \in \mathcal{S}_A \;\wedge\; t_{B\bar{B}} \in \mathcal{S}_B \right\} \tag{3.1}$$

Each transformation that belongs to the set $\mathcal{L}_{AB}$ equivalently describes the relative location of the features. Since geometric relations can be derived from this relative location, the most simple transformation belonging to $\mathcal{L}_{AB}$ (the one that has the minimum possible number of translations and rotations) contains one translation or rotation corresponding to each of the parameters defining the geometric relations between the elements. This means that there exist references $\bar{A}$ and $\bar{B}$ where the geometric relations between the two elements are linear functions of $\mathbf{x}_{\bar{A}\bar{B}}$. In the above example $\mathbf{x}_{\bar{A}\bar{B}}$ contains one translation and one rotation corresponding to the distance and angle between the edges. This transformation is then, the simplest transformation $t_{\bar{A}\bar{B}}$ that satisfies (3.1).

Our purpose is to describe the general form of $\mathbf{x}_{A\bar{A}}$, $\mathbf{x}_{B\bar{B}}$, $\mathbf{x}_{\bar{A}\bar{B}}$, for all types of geometric features. We will show that they always exist and in the general case are uniquely determined, up to common symmetries. In the following we describe the general procedure to determine the aligning transformations for all combinations of subgroups of symmetries.

Determination of the Aligning Transformations

Let A and B represent the location of two geometric features. Let $\mathcal{S}_A$ and $\mathcal{S}_B$ be the subgroups of transformations of the corresponding features, and let $t_{A\bar{A}} \in \mathcal{S}_A$, and $t_{B\bar{B}} \in \mathcal{S}_B$. Let the relative location vector between the features be $\mathbf{x}_{AB} = (x,\ y,\ z,\ \psi,\ \theta,\ \phi)^T$. The general form of $\mathbf{x}_{A\bar{A}}$ and $\mathbf{x}_{B\bar{B}}$

depend on the subgroup of symmetries to which they belong. $\mathbf{x}_{\bar{A}\bar{B}}$ can be easily calculated from:

$$\mathbf{x}_{\bar{A}\bar{B}} = \ominus \mathbf{x}_{A\bar{A}} \oplus \mathbf{x}_{AB} \oplus \mathbf{x}_{B\bar{B}} \tag{3.2}$$

Each component of the location vector $\mathbf{x}_{\bar{A}\bar{B}} = (\bar{x},\ \bar{y},\ \bar{z},\ \bar{\psi},\ \bar{\theta},\ \bar{\phi})^T$ is a function of the components of the location vectors $\mathbf{x}_{A\bar{A}}$ and $\mathbf{x}_{B\bar{B}}$. The aligning transformations are obtained by determining the values of the components of $\mathbf{x}_{A\bar{A}}$ and $\mathbf{x}_{B\bar{B}}$ such that the greatest number of components of $\mathbf{x}_{\bar{A}\bar{B}}$ are *anullated.* This process is best understood by means of an example.

Example 3.2: *Aligning transformations for two edges*
Let A and B represent the location of two edges. The symmetries of edges are $T_x\, R_x$, which means $\mathcal{S}_A = \mathcal{S}_B = T_x\, R_x$. Thus we take:

$$\begin{aligned} \mathbf{x}_{A\bar{A}} &= (x_a,\ 0,\ 0,\ \psi_a,\ 0,\ 0)^T \\ \mathbf{x}_{B\bar{B}} &= (x_b,\ 0,\ 0,\ \psi_b,\ 0,\ 0)^T \end{aligned}$$

Our purpose is to find the values of x_a, x_b, ψ_a and ψ_b such that $\mathbf{x}_{\bar{A}\bar{B}}$ contains the minimum number of translations and rotations. From (3.2) we have that the components of $\mathbf{x}_{\bar{A}\bar{B}}$ are given by:

$$\bar{x} = x - x_a + x_b\ \cos\theta\cos\phi \tag{3.3}$$

$$\bar{y} = y\ \cos\psi_a + z\ \sin\psi_a + x_b\ (\cos\psi_a\sin\phi\cos\theta - \sin\psi_a\sin\theta) \tag{3.4}$$

$$\bar{z} = -y\ \sin\psi_a + z\ \cos\psi_a - x_b\ (\sin\psi_a\sin\phi\cos\theta + \cos\psi_a\sin\theta) \tag{3.5}$$

$$\bar{\psi} = \tan^{-1}\left(\frac{\begin{array}{r} -\left(\cos\phi\cos(\psi+\psi_b)+\sin\phi\sin\theta\sin(\psi+\psi_b)\right)\sin\psi_a \\ +\sin(\psi+\psi_b)\cos\psi_a\cos\theta \end{array}}{\begin{array}{r} (\cos\phi\sin(\psi+\psi_b)-\sin\phi\sin\theta\cos(\psi+\psi_b))\sin\psi_a \\ +\cos(\psi+\psi_b)\cos\psi_a\cos\theta \end{array}}\right) \tag{3.6}$$

$$\bar{\theta} = \tan^{-1}\left(\frac{\sin\psi_a\sin\phi\cos\theta+\cos\psi_a\sin\theta}{\sqrt{\cos^2\theta\cos^2\phi + (\cos\psi_a\sin\phi\cos\theta - \sin\psi_a\sin\theta)^2}}\right) \tag{3.7}$$

$$\bar{\phi} = \tan^{-1}\left(\frac{\cos\psi_a\sin\phi\cos\theta - \sin\psi_a\sin\theta}{\cos\phi\cos\theta}\right) \tag{3.8}$$

To solve this problem, we make the following considerations:

1. There are four free variables: x_a, x_b, ψ_a, and ψ_b. Thus, in the general case we should be able to anullate four components of $\mathbf{x}_{\bar{A}\bar{B}}$: two distances and two angles.

2. Equation (3.3) is the only one that contains x_a. Thus, we use this variable to anullate $\bar{x}$. We do this by taking:

$$x_a = x - x_b \cos\theta \cos\phi$$

3. We can use x_b to anullate $\bar{y}$ or $\bar{z}$. We choose $\bar{y}$ (the alternative choice would yield a completely equivalent solution). For this, we take x_b as:

$$x_b = -\frac{y \cos\psi_a + z \sin\psi_a}{\cos\psi_a \sin\phi\cos\theta - \sin\psi_a \sin\theta}$$

4. Equations (3.7) and (3.8) contain ψ_a. We use ψ_a to anullate $\bar{\theta}$:

$$\psi_a = -\tan^{-1}\left(\frac{\sin\theta}{\sin\phi\cos\theta}\right)$$

5. ψ_b can be used to anullate the value of $\bar{\psi}$:

$$\psi_b = \tan^{-1}\left(\frac{\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi}{\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi}\right)$$

This solution makes zero all components of $\mathbf{x}_{\bar{A}\bar{B}}$, but $\bar{z}$ and $\bar{\phi}$. We can calculate their value from (3.5) and (3.8):

$$\begin{aligned}
\bar{z} &= \frac{y\sin\theta + z\sin\phi\cos\theta}{\sqrt{1-\cos\phi^2\cos\theta^2}} \\
\bar{\phi} &= \tan^{-1}\left(\frac{\sqrt{1-\cos\phi^2\cos\theta^2}}{\cos\phi\cos\theta}\right)
\end{aligned}$$

This coincides precisely with the expected results. Thus, in the general case, a distance and an angle are the geometric relations between two edges.

Particular cases: note that this solution is not valid when $\theta = \phi = 0$. This corresponds to the case where the two edges are parallel. In this case, the components of $\mathbf{x}_{\bar{A}\bar{B}}$ become:

$$\begin{aligned}
\bar{x} &= x + x_b - x_a & (3.9)\\
\bar{y} &= y\cos\psi_a + z\sin\psi_a & (3.10)\\
\bar{z} &= z\cos\psi_a - y\sin\psi_a & (3.11)\\
\bar{\psi} &= \frac{\cos(\psi+\psi_b)\sin\psi_a - \sin(\psi+\psi_b)\cos\psi_a}{-\sin(\psi+\psi_b)\sin\psi_a - \cos(\psi+\psi_b)\cos\psi_a} & (3.12)\\
\bar{\theta} &= 0 & (3.13)\\
\bar{\phi} &= 0 & (3.14)
\end{aligned}$$

We obtain a solution in the following way:

1. Since there is only one remaining angle and two free angular variables, we can use one to anullate a distance. (Note that the contrary is not possible, to anullate an angular component with a translation variable.) We choose to anullate $\bar{y}$ in (3.10) with ψ_a:

$$\psi_a = -\tan^{-1}\left(\frac{y}{z}\right)$$

2. The angle ψ can be anullated in (3.12) with the remaining angular variable ψ_b:

$$\psi_b = \tan^{-1}\left(\frac{y\cos\psi + z\sin\psi}{y\sin\psi - z\cos\psi}\right)$$

3. The only remaining variable that can be anullated is $\bar{x}$. We can use x_a to anullate its value in (3.9):

$$x_a = x + x_b$$

In this case, there only remains one non-zero variable in $\mathbf{x}_{\bar{A}\bar{B}}$:

$$\bar{z} = \sqrt{y^2 + z^2}$$

This indicates that for parallel edges only a distance can be measured. Note that the value of x_b remains free. This is because the distance between parallel edges can be measured at any point along them.

Again, this solution is not valid when $y = z = \theta = \phi = 0$. In this case the edges are coincident, so the components of $\mathbf{x}_{\bar{A}\bar{B}}$ become:

$$\begin{aligned}
\bar{x} &= x - x_a + x_b \\
\bar{y} &= 0 \\
\bar{z} &= 0 \\
\bar{\psi} &= \psi - \psi_a + \psi_b \\
\bar{\theta} &= 0 \\
\bar{\phi} &= 0
\end{aligned}$$

It can be seen that both $\bar{x}$ and $\bar{\psi}$ can be easily anullated, still remaining two free variables:

$$\begin{aligned}
x_a &= x + x_b \\
\psi_a &= \psi_b + \psi \\
x_b & \quad \text{free} \\
\psi_b & \quad \text{free}
\end{aligned}$$

These particular cases, where there remain free variables in the solutions, correspond to feature locations that do not allow to determine the location of the object. Therefore, features whose geometric relations belong to one of these particular cases are avoided during the initial steps of recognition (chapter 6). ◇

The solutions for all the combinations of symmetries are obtained in a similar way and are summarized in table 3.1. In this table we can find the general form of vector $\mathbf{x}_{\bar{A}\bar{B}}$. The form that this vector adopts for some particular cases also appears. For example, given two edges ($\mathcal{S}_A = \mathcal{S}_B = T_x \, R_x$), in case they are parallel ($\theta = \phi = 0$), only a distance ($t_{\bar{z}}$) can be measured. We do not consider features whose relative location is near these particular cases , because they correspond to situations where the pair of observations do not allow to determine the object location. A detailed description of each solution can be found in appendix B.

In summary, given two geometric features A and B, whose subgroups of symmetries are $\mathcal{S}_A$ and $\mathcal{S}_B$, the location vector corresponding to the geometric relations between the two elements is computed following these steps:

1. Using $\mathcal{S}_A$ and $\mathcal{S}_B$ and the values of the components of $\mathbf{x}_{AB}$, from the corresponding table (appendix B) obtain the expressions corresponding to $\mathbf{x}_{A\bar{A}}$ and $\mathbf{x}_{B\bar{B}}$, and calculate the values of their components.

2. Calculate $\mathbf{x}_{\bar{A}\bar{B}}$ using (3.2).

3. Some components of this location vector correspond to the geometric relations $\mathbf{r}$ between the features and the rest are zero. We use a row-selection matrix S_{AB} to extract them (the value of S_{AB} depends on the general form of $\mathbf{x}_{\bar{A}\bar{B}}$, described in table 3.1):

$$\mathbf{r} = S_{AB} \, \mathbf{x}_{\bar{A}\bar{B}}$$

Vector $\mathbf{r}$ will contain the value of the geometric relations between features A and B. Function `calculate_geometric_relations` is implemented using this procedure. This function is used to determine the values for the geometric relations between two geometric elements given their relative location $\mathbf{x}_{AB}$, and their respective subgroups of symmetries, $\mathcal{S}_A$ and $\mathcal{S}_B$ (see algorithm 3.1). This will allow to determine whether the geometric relations between two model features are satisfied by the geometric relations between two observations.

$\mathcal{S}_A$	$\mathcal{S}_B$	Case of t_{AB}	General Form of $t_{\bar{A}\bar{B}}$
R_x	R_x		$t_{\bar{x}} \cdot t_{\bar{y}} \cdot t_{\bar{z}} \cdot r_{\bar{\phi}}$
		$\theta = \phi = 0$	$t_{\bar{x}} \cdot t_{\bar{y}}$
		$y = z = \theta = \phi = 0$	$t_{\bar{x}}$
R_x	R_{xyz}		$t_{\bar{x}} \cdot t_{\bar{z}}$
		$y = z = 0$	$t_{\bar{x}}$
T_x	T_x		$t_{\bar{y}} \cdot r_{\bar{\phi}} \cdot r_{\bar{\theta}} \cdot r_{\bar{\psi}}$
		$\theta = 0$	$t_{\bar{z}} \cdot r_{\bar{\phi}} \cdot r_{\bar{\psi}}$
		$\theta = \phi = 0$	$t_{\bar{y}} \cdot t_{\bar{z}} \cdot r_{\bar{\psi}}$
T_x	R_x		$t_{\bar{y}} \cdot t_{\bar{z}} \cdot r_{\bar{\phi}} \cdot r_{\bar{\psi}}$
T_x	$T_x\, R_x$		$t_{\bar{y}} \cdot r_{\bar{\phi}} \cdot r_{\bar{\psi}}$
		$\theta = \phi = 0$	$t_{\bar{y}} \cdot t_{\bar{z}}$
T_x	R_{xyz}		$t_{\bar{y}} \cdot t_{\bar{z}}$
$T_x\, R_x$	R_x		$t_{\bar{y}} \cdot t_{\bar{z}} \cdot r_{\bar{\phi}}$
		$\theta = \phi = 0$	$t_{\bar{z}}$
$T_x\, R_x$	$T_x\, R_x$		$t_{\bar{z}} \cdot r_{\bar{\phi}}$
		$\theta = \phi = 0$	$t_{\bar{z}}$
$T_x\, R_x$	R_{xyz}		$t_{\bar{z}}$
$T_{xy}\, R_z$	R_x		$t_{\bar{z}} \cdot r_{\bar{\theta}}$
$T_{xy}\, R_z$	R_{xyz}		$t_{\bar{z}}$
$T_{xy}\, R_z$	T_x		$r_{\bar{\theta}} \cdot r_{\bar{\psi}}$
		$\theta = 0$	$t_{\bar{z}} \cdot r_{\bar{\psi}}$
$T_{xy}\, R_z$	$T_x\, R_x$		$r_{\bar{\theta}}$
		$\theta = 0$	$t_{\bar{z}}$
$T_{xy}\, R_z$	$T_{xy}\, R_z$		$r_{\bar{\theta}}$
		$\theta = \psi = 0$	$t_{\bar{z}}$
R_{xyz}	R_{xyz}		$t_{\bar{z}}$

$$\begin{aligned} t_{\bar{x}} &= \text{Trans}(\bar{x}, 0, 0); \quad r_{\bar{\psi}} = \text{Rot}(x, \bar{\psi}) \\ t_{\bar{y}} &= \text{Trans}(0, \bar{y}, 0); \quad r_{\bar{\theta}} = \text{Rot}(y, \bar{\theta}) \\ t_{\bar{z}} &= \text{Trans}(0, 0, \bar{z}); \quad r_{\bar{\phi}} = \text{Rot}(z, \bar{\phi}) \end{aligned}$$

Table 3.1: Geometric relations for all combinations of symmetries of geometric elements

Estimating Uncertain Geometric Relations

Let us consider now the estimation of geometric relations between uncertain observations of features. Given two geometric features with associated references A and B, whose uncertain locations are expressed by $\mathbf{L}_{WA} = (\hat{\mathbf{x}}_{WA}, \hat{\mathbf{p}}_A, C_A)$ and $\mathbf{L}_{WB} = (\hat{\mathbf{x}}_{WB}, \hat{\mathbf{p}}_B, C_B)$ respectively, the geometric relations between them can be calculated by estimating the values of the components of the location vector $\mathbf{x}_{\bar{A}\bar{B}}$ and its covariance. In order to reduce errors due to the use of a linear approximation, we will consider the estimations centered ($\hat{\mathbf{p}}_A = \hat{\mathbf{p}}_B = 0$). The procedure is the following:

1. Calculate the location vectors $\mathbf{x}_{A\bar{A}}$ and $\mathbf{x}_{B\bar{B}}$ using the table in appendix B corresponding to $\mathcal{S}_A$ and $\mathcal{S}_B$. These location vectors can be used to calculate the uncertain location of $\mathbf{L}_{W\bar{A}}$, as follows:

$$\mathbf{L}_{W\bar{A}} = \mathbf{L}_{WA} \oplus \mathbf{x}_{A\bar{A}}$$

According to (2.3), and given that $B_{\bar{A}} = B_A s$, the result will be $\mathbf{L}_{W\bar{A}} = (\hat{\mathbf{x}}_{W\bar{A}}, \hat{\mathbf{p}}_{\bar{A}}, C_{\bar{A}})$, where:

$$\begin{aligned} \hat{\mathbf{x}}_{W\bar{A}} &= \hat{\mathbf{x}}_{WA} \oplus \mathbf{x}_{A\bar{A}} \\ \hat{\mathbf{p}}_{\bar{A}} &= B_A J_{\bar{A}A} B_A^T \, \hat{\mathbf{p}}_A \; = 0 \\ C_{\bar{A}} &= B_A J_{\bar{A}A} B_A^T \; C_A \; B_A J_{\bar{A}A}^T B_A^T \end{aligned} \tag{3.15}$$

Matrix $J_{\bar{A}A}$ is the Jacobian of the transformation represented by $\mathbf{x}_{A\bar{A}}$ (see appendix A). The uncertain location of $\bar{B}$, $\mathbf{L}_{WB} = (\hat{\mathbf{x}}_{W\bar{B}}, \hat{\mathbf{p}}_{\bar{B}}, C_{\bar{B}})$, is calculated in the same way.

2. The relative location vector $\mathbf{x}_{\bar{A}\bar{B}}$ is given by:

$$\begin{aligned} \mathbf{x}_{\bar{A}\bar{B}} &= \ominus \mathbf{x}_{W\bar{A}} \oplus \mathbf{x}_{W\bar{B}} \\ &= \ominus(\mathbf{x}_{W\bar{A}} \oplus B_A^T \mathbf{p}_{\bar{A}}) \oplus (\mathbf{x}_{W\bar{B}} \oplus B_B^T \mathbf{p}_{\bar{B}}) \\ &= \ominus \mathbf{d}_{\bar{A}} \ominus \hat{\mathbf{x}}_{W\bar{A}} \oplus \hat{\mathbf{x}}_{W\bar{B}} \oplus \mathbf{d}_{\bar{B}} \\ &= \ominus \mathbf{d}_{\bar{A}} \oplus \hat{\mathbf{x}}_{\bar{A}\bar{B}} \oplus \mathbf{d}_{\bar{B}} \end{aligned}$$

Thus, we can calculate $\hat{\mathbf{x}}_{\bar{A}\bar{B}}$ and its covariance as follows:

$$
\begin{aligned}
\hat{\mathbf{x}}_{\bar{A}\bar{B}} &= \ominus \hat{\mathbf{x}}_{W\bar{A}} \oplus \hat{\mathbf{x}}_{W\bar{B}} \\
Cov(\mathbf{x}_{\bar{A}\bar{B}}) &= J_{1\oplus}\{0, \hat{\mathbf{x}}_{\bar{A}\bar{B}}\} B_A^T C_{\bar{A}} B_A J_{1\oplus}^T\{0, \hat{\mathbf{x}}_{\bar{A}\bar{B}}\} \\
&+ J_{2\oplus}\{\hat{\mathbf{x}}_{\bar{A}\bar{B}}, 0\} B_B^T C_{\bar{B}} B_B J_{2\oplus}^T\{\hat{\mathbf{x}}_{\bar{A}\bar{B}}, 0\} \qquad (3.16)
\end{aligned}
$$

where $J_{1\oplus}$ and $J_{2\oplus}$ are the Jacobians of the composition of location vectors (see appendix A).

3. To select the elements of this location vector which correspond to the geometric relations $\mathbf{r}$ between the features, we premultiply by the row-selection matrix S_{AB}, whose values depend on $\mathcal{S}_A$ and $\mathcal{S}_B$, and can be easily deduced from table 3.1:

$$
\begin{aligned}
\hat{\mathbf{r}} &= S_{AB}\, \hat{\mathbf{x}}_{\bar{A}\bar{B}} \\
Cov\,(\mathbf{r}) &= S_{AB}\, Cov(\mathbf{x}_{\bar{A}\bar{B}})\, S_{AB}^T \qquad (3.17)
\end{aligned}
$$

Vector $\hat{\mathbf{r}}$ and its covariance constitute an estimation of the geometric relations between A and B.

In summary, the estimation of the geometric relations between two observed features can be made with function `estimate_geometric_relations`, shown in algorithm 3.1.

Example 3.3: *Estimated geometric relations between edges*
Suppose we use a proximity sensor to observe two edges of equal length l, measuring n equally distant points for each, we associate to them references A and B. Supposing the sensor's measurement precision is equal to σ in all directions, from (2.16), we have that their covariance matrices would be:

$$
C_A = C_B = \frac{\sigma^2}{n}\,\mathrm{diag}\left(1, 1, \frac{12}{l^2}, \frac{12}{l^2}\right)
$$

Recall from example 3.2.1 that the aligning transformations for two edges are $\mathbf{x}_{A\bar{A}} = (x_a,\ 0,\ 0,\ \psi_a,\ 0,\ 0)^T$, $\mathbf{x}_{B\bar{B}} = (x_b,\ 0,\ 0,\ \psi_b,\ 0,\ 0)^T$, and $\mathbf{x}_{\bar{A}\bar{B}} = (0,\ 0,\ \bar{z},\ 0,\ 0,\ \bar{\phi})^T$. From (3.15) we have:

$$
C_{\bar{A}} = \frac{\sigma^2}{n\, l^2}
\begin{bmatrix}
l^2 + 12\, x_a^2 & 0 & 0 & 12\, x_a \\
0 & l^2 + 12\, x_a^2 & -12\, x_a & 0 \\
0 & -12\, x_a & 12 & 0 \\
12\, x_a & 0 & 0 & 12
\end{bmatrix}
$$

$$C_{\bar{B}} \quad = \quad \frac{\sigma^2}{n\,l^2} \begin{bmatrix} l^2 + 12\,x_b^2 & 0 & 0 & 12\,x_b \\ 0 & l^2 + 12\,x_b^2 & -12\,x_b & 0 \\ 0 & -12\,x_b & 12 & 0 \\ 12\,x_b & 0 & 0 & 12 \end{bmatrix}$$

The matrix S_{AB} which selects the desired components of the location vector is:

$$S_{AB} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The covariance of $\mathbf{x}_{\bar{A}\bar{B}}$ can be calculated from (3.16) and (3.17), giving:

$$Cov(\mathbf{r}) = 2\,\sigma^2 \begin{bmatrix} \frac{6\left(x_a^2 + x_b^2\right)}{l^2} + \frac{1}{n} & 0 \\ 0 & \frac{12}{l^2} \end{bmatrix}$$

We can see that the uncertainty in the estimation of the distance between the two edges is related to how far the edges are from the common perpendicular (described by x_a, and x_b). This is due to the fact that the uncertainty in the location of the edge point where the common perpendicular is located is coupled with the orientation uncertainty of the edge, so that a deviation in edge orientation produces a deviation in the position of this point. In this case, uncertainty in the angle between the edges does not depend on the values of $\mathbf{x}_{A\bar{A}}$ or $\mathbf{x}_{B\bar{B}}$. ◇

The preceding example illustrates the fact that the precision in the estimation of geometric relations depends on the relative location between the features. In general, the estimation becomes less precise when the features are more distant from each other, or when their relative location is close to a singular case. This information is useful in determining how discriminant the constraint validation mechanism will be, and thus, will be useful during the recognition process.

Computational Considerations

In applying this procedure to compute the necessary transformations, there are several things that must be taken into account:

1. *For any pair of features, it is irrelevant which one you take as A, and which as B.* Nevertheless, one solution may be computationally less expensive than the other.

   Example 3.4: *Geometric relations between a dihedral and a vertex*
   If we associate reference A to the dihedral, we have $\mathcal{S}_A = T_x$ and $\mathcal{S}_B = R_{xyz}$. Thus:

$$\begin{aligned}\mathbf{x}_{A\bar{A}} &= (x_a,\ 0,\ 0,\ 0,\ 0,\ 0)^T\\ \mathbf{x}_{B\bar{B}} &= (0,\ 0,\ 0,\ \psi_b,\ \theta_b,\ \phi_b)^T\end{aligned}$$

The solution in this case yields that $\mathbf{x}_{\bar{A}\bar{B}} = (0,\ \bar{y},\ \bar{z},\ 0,\ 0,\ 0)^T$ when:

$$\begin{aligned}x_a &= x\\ \psi_a &= \tan^{-1}\left(\frac{\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi}{\cos\theta\cos\psi}\right)\\ \theta_a &= -\tan^{-1}\left(\frac{\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi}{\sqrt{\cos^2\theta\cos^2\phi + (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)^2}}\right)\\ \phi_a &= -\tan^{-1}\left(\frac{\sin\phi\cos\psi - \cos\phi\sin\theta\sin\psi}{\cos\phi\cos\theta}\right)\end{aligned}$$

On the other hand, if we associate reference A to the vertex, we would have:

$$\begin{aligned}\mathbf{x}_{A\bar{A}} &= (0,\ 0,\ 0,\ \psi_a,\ \theta_a,\ \phi_a)^T\\ \mathbf{x}_{B\bar{B}} &= (x_b,\ 0,\ 0,\ 0,\ 0,\ 0)^T\end{aligned}$$

This would yield the following solution:

$$\begin{aligned}x_b &= -x\cos\theta\cos\phi - y\cos\theta\sin\phi + z\sin\theta\\ \psi_a &= \psi\\ \theta_a &= \theta\\ \phi_a &= \phi\\ \mathbf{x}_{\bar{A}\bar{B}} &= (0,\ \bar{y},\ \bar{z},\ 0,\ 0,\ 0)^T\end{aligned}$$

◇

In general, simpler solutions are preferred.

2. *RPY angles have a singularity when $\theta = \pm\pi/2$.* The SPmodel represents orientations using RPY angles. This means that when θ is near this value, its covariance is very high because a small perturbation may

cause a great change in its value. This is reflected in the Jacobian matrices, where it originates a division by zero (see appendix A), and thus, an infinite covariance. Thus, the case when the geometric relations between two features give this singular solution must be avoided. The adopted solution consists in rotating the reference associated to the one of the features so that the value of the $\bar{\theta}$ component of their relative location vector goes away from the singularity.

Example 3.5: *Geometric relations between dihedrals*
Consider two geometric features corresponding to dihedrals ($\mathcal{S} = T_x$). Since dihedrals have no symmetries of rotation, it is not possible to anullate any angular component of the relative location vector, and it may be the case that $\bar{\theta}$ approaches $\pm\pi/2$. This can be solved by rotating the reference associated to the first dihedral so that the value of $\bar{\theta}$ changes from $\pm\pi/2$ from 0. This can be done by modifying (3.2) in the following way:

$$\mathbf{x}_{\bar{A}\bar{B}} = \ominus\mathbf{x}_{A\bar{A}} \ominus \mathrm{Rot}(x, -\pi/2) \oplus \mathbf{x}_{AB} \oplus \mathbf{x}_{B\bar{B}}$$

This implies altering the convention that the z axes of the reference associated to a dihedral bisects the angle between the two planes that form the dihedral. In performing this alteration, it will be the y axes the one who bisects the angle. ◇

Note that this alteration is only necessary for computing geometric relations and should be done both in the observations and in the model features to be paired with them.

3.2.2 Validating Binary Relations

We use the results of the preceding section to define a procedure to validate the geometric relations between two observed features and their corresponding model features of an object. Given a vector $\mathbf{r}_m$, which contains the value of the geometric relations between the model features an the object, and $\hat{\mathbf{r}}_o$, $Cov(\mathbf{r}_o)$, the estimation of the geometric relations between the observed features, we can measure the discrepancy between $\mathbf{r}_m$ and $\mathbf{r}_o$ using the *Mahalanobis distance* [Cuadras 89]:

$$D^2 = (\hat{\mathbf{r}}_o - \mathbf{r}_m)^T Cov(\mathbf{r}_o)^{-1}(\hat{\mathbf{r}}_o - \mathbf{r}_m) \tag{3.18}$$

Under the Gaussianity hypothesis, D^2 follows a chi-square distribution with $m = \dim(\mathbf{r}_m)$ degrees of freedom. For a given significance level α, $\mathbf{r}_o$ can be considered compatible with $\mathbf{r}_m$ if:

```
FUNCTION satisfy_binary_constraints ($p_A$, $p_B$)

; $p_A = (e_A, m_A)$ and $p_B = (e_B, m_B)$: pairings between observations and model features
; Determines whether the estimated geometric relations between observations
; $e_A$ and $e_B$ can be considered compatible
; with the geometric relations between model features $m_A$ and $m_B$
; returns TRUE if so, FALSE otherwise

    $\mathbf{r}_m$ := calulate_geometric_relations ($m_A$, $m_B$);
    $\mathbf{r}_o, C_o$ := estimate_geometric_relations ($e_A$, $e_B$);
    $D^2 := (\mathbf{r}_o - \mathbf{r}_m)^T C_o^{-1} (\mathbf{r}_o - \mathbf{r}_m)$;
    $D_m^2$ := chi_square ($\dim(\mathbf{r}_m)$, $\alpha$);

RETURN $D^2 \leq D_m^2$;
END;


FUNCTION calculate_geometric_relations ($m_A$, $m_B$)

; $m_A = (\mathbf{x}_{OA}, \mathcal{S}_A)$ and $m_B = (\mathbf{x}_{OB}, \mathcal{S}_B)$: model features
; $\mathbf{x}_{OA}$ and $\mathbf{x}_{OB}$: location vectors representing the location of the features relative to the object
; $\mathcal{S}_A$ and $\mathcal{S}_B$: subgroups of symmetries of the features
; Calculates the geometric relations $\mathbf{r}$ between two model features $A$ and $B$
; obtaining from table 3.1 the appropriate
; aligning transformation for symmetries of $A$ and $B$

    $\mathbf{x}_{AB} := \ominus \mathbf{x}_{OA} \oplus \mathbf{x}_{OB}$;
    $\mathbf{x}_{A\bar{A}}, \mathbf{x}_{B\bar{B}}$ := calculate_from_table ($\mathbf{x}_{AB}$, $\mathcal{S}_A$, $\mathcal{S}_B$);
    $\mathbf{x}_{\bar{A}\bar{B}} := \ominus \mathbf{x}_{A\bar{A}} \oplus \mathbf{x}_{AB} \oplus \mathbf{x}_{B\bar{B}}$;
    $\mathbf{r} := S_{AB}\ \mathbf{x}_{\bar{A}\bar{B}}$;

RETURN $\mathbf{r}$;
END;


FUNCTION estimate_geometric_relations ($e_A$, $e_B$)

; $e_A = (\mathbf{L}_{WA}, \mathcal{S}_A)$ and $e_B = (\mathbf{L}_{WB}, \mathcal{S}_B)$: observations
; $\mathbf{L}_{WA} = (\hat{\mathbf{x}}_{WA}, \hat{\mathbf{p}}_A, C_A)$ and $\mathbf{L}_{WB} = (\hat{\mathbf{x}}_{WB}, \hat{\mathbf{p}}_B, C_B)$: uncertain locations
; of the features in the scene
; $\mathcal{S}_A$ and $\mathcal{S}_B$: subgroups of symmetries of the features
; Calculates the estimated value $\hat{\mathbf{r}}$ and covariance $C_{\hat{\mathbf{r}}}$ of the geometric relations
; between two observed features $A$ and $B$, obtaining from table 3.1 the appropriate
; aligning transformation for the symmetries of $A$ and $B$

    $\hat{\mathbf{x}}_{AB} := \ominus \hat{\mathbf{x}}_{WA} \oplus \hat{\mathbf{x}}_{WB}$;
    $\hat{\mathbf{x}}_{A\bar{A}}, \hat{\mathbf{x}}_{B\bar{B}}$ := calculate_from_table ($\hat{\mathbf{x}}_{AB}$, $\mathcal{S}_A$, $\mathcal{S}_B$);
    $\hat{\mathbf{x}}_{\bar{A}\bar{B}} := \ominus \hat{\mathbf{x}}_{A\bar{A}} \oplus \hat{\mathbf{x}}_{AB} \oplus \hat{\mathbf{x}}_{B\bar{B}}$;
    $\hat{\mathbf{r}} := S_{AB}\ \hat{\mathbf{x}}_{\bar{A}\bar{B}}$;
    $C_{\bar{A}\bar{B}} := J_{1\oplus}\{0, \hat{\mathbf{x}}_{\bar{A}\bar{B}}\} B_A^T C_{\bar{A}} B_A J_{1\oplus}^T \{0, \hat{\mathbf{x}}_{\bar{A}\bar{B}}\} + J_{2\oplus}\{\hat{\mathbf{x}}_{\bar{A}\bar{B}}, 0\} B_B^T C_{\bar{B}} B_B J_{2\oplus}^T \{\hat{\mathbf{x}}_{\bar{A}\bar{B}}, 0\}$;
    $C_{\hat{\mathbf{r}}} := S_{AB}\ C_{\mathbf{x}_{\bar{A}\bar{B}}}\ S_{AB}^T$;

RETURN $\hat{\mathbf{r}}$, $C_{\hat{\mathbf{r}}}$;
END;
```

Algorithm 3.1: *Validating Geometric Constraints*

$$D^2 \leq D^2_{m,\alpha} \tag{3.19}$$

where $D^2_{m,\alpha}$ is a threshold value, obtained from the χ^2_m distribution, such that the probability of rejecting a good matching is α. This constraint validation method use used to implement function `satisfy_binary_constraints` (algorithm 3.1).

Example 3.6: *Constraint validation for two edges*
Continuing with our example, let $\Delta\mathbf{r} = \hat{\mathbf{r}}_o - \mathbf{r}_m = (\Delta z, \Delta\phi)^T$ be the discrepancy between the estimated value of the geometric relations of the observed edges and those of the object model. According to (3.18) we have:

$$D^2 = \frac{n\,l^2}{2\,\sigma^2}\left(\frac{(\Delta z)^2}{6\,x_b^2 + 6\,x_a^2 + l^2} + \frac{(\Delta\phi)^2}{12}\right)$$

For a given significance level α, we have that the observed edges are considered compatible with the model edges if $D^2 \leq D^2_{m,\alpha}$. We can see that the value of D^2 increases with n, which means that longer edges will allow the test to be more discriminant. On the other hand, D^2 decreases with σ^2, because less precise sensors decrease the selectivity of the test. D^2 also decreases with x_a and x_b, the relative distances of the edge references to their common perpendicular. These values can be large for two reasons:

1. The observed edges are very distant from one another. The effect of the orientation errors of the edge on the position errors of points on the edge grow with the distance between the point and the edge reference.
2. The relative location of the edges is close to a singular case. The singular case for edges is when they are parallel.

The confirms the intuitive idea that distant edges, or edges whose relative location is close to a singular case, decrease the discriminant power of the constraint validation mechanism. ◇

3.3 Location Dependent Geometric Relations

In the previous section we studied geometric relations whose values do not depend on the object location. They are useful to validate the consistency of a set of observations with respect to a set of model features, without the need to estimate the location of the object. However, the availability of an estimation of the location of the object gives us the possibility of applying some other validation mechanisms on the observations. In this section we describe the most important location dependent constraints.

3.3.1 The Rigidity Constraint

The fundamental location dependent constraint is denominated *rigidity*. Intuitively, rigidity states that the location of an object determines the location of its features in the scene. Given an estimation $\mathbf{L}_{WO} = (\hat{\mathbf{x}}_{WO}, \hat{\mathbf{d}}_O, C_O)$ of the location of an object, and given the relative location of the feature with respect to the object model, $\mathbf{x}_{OM}$, we can estimate the location of the feature in the scene as follows:

$$\mathbf{L}_{WM} = \mathbf{L}_{WO} \oplus \mathbf{x}_{OM}$$

According to (2.4), the resulting uncertain location of M is given by $\mathbf{L}_{WM} = (\hat{\mathbf{x}}_{WM}, \hat{\mathbf{p}}_M, C_M)$, where:

$$\begin{aligned} \hat{\mathbf{x}}_{WM} &= \hat{\mathbf{x}}_{WM} \oplus \mathbf{x}_{OM} \\ \hat{\mathbf{p}}_M &= B_M \, J_{MO} \, \hat{\mathbf{d}}_O \\ C_M &= B_M J_{MO} C_O J_{MO}^T B_M^T \end{aligned} \tag{3.20}$$

The test of whether an observation is compatible with M can be applied both to observed features and subfeatures. Both cases are explained next.

Rigidity for Features

Given an observed feature whose estimated location in the scene is represented by $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E, C_E)$, E can be considered compatible with M if their location coincide, up to symmetries. This condition can be expressed by the *inverse constraint*:

$$B_E \mathbf{x}_{EM} = 0$$

where B_E is the self-binding matrix of the feature. We can measure the discrepancy between the model feature and the feature observed in the scene using the Mahalanobis distance:

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

Distance D^2 follows a chi-square distribution with $m = \dim(\mathbf{p}_E) = \text{rank}(B_E)$ degrees of freedom. The estimated value of $\mathbf{x}_{EM}$ and its covariance can be obtained in the following way:

$$\begin{aligned} \hat{\mathbf{x}}_{EM} &= \ominus\hat{\mathbf{x}}_{WE} \oplus \hat{\mathbf{x}}_{WM} \\ Cov(\mathbf{x}_{EM}) &= J_{1\oplus}\{0, \hat{\mathbf{x}}_{EM}\} B_A^T C_E B_E J_{1\oplus}^T\{0, \hat{\mathbf{x}}_{EM}\} \\ &+ J_{2\oplus}\{\hat{\mathbf{x}}_{EM}, 0\} B_M^T C_M B_M J_{2\oplus}^T\{\hat{\mathbf{x}}_{EM}, 0\} \end{aligned} \tag{3.21}$$

We use a hypothesis test similar to (3.19): if the estimated distance is greater than the threshold $D^2_{m,\alpha}$, the compatibility hypothesis is rejected.

Rigidity for Subfeatures

Let us now consider the case where the observation is a subfeature, whose estimated location in the scene is represented by $\mathbf{L}_{WP} = (\hat{\mathbf{x}}_{WP}, \hat{\mathbf{p}}_P, C_P)$. Given a model feature M, we can consider that P is compatible with M if their relative location satisfies either the *direct* or *inverse constraint*:

$$\begin{aligned} B_{MP}\mathbf{x}_{MP} &= 0 \\ B_{PM}\mathbf{x}_{PM} &= 0 \end{aligned}$$

where B_{MP} and B_{PM} are the binding matrices of the pairing, depending on how this pairing can be expressed. Let us consider the direct constraint. The discrepancy between the model feature and the subfeature observed in the scene is measured using the Mahalanobis distance:

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

In this case, distance D^2 follows a chi-square distribution with $m = \text{rank}(B_{MP})$ degrees of freedom. The estimated value of $\mathbf{x}_{MP}$ and its covariance can be obtained using (3.16). A hypothesis test similar to that of the preceding paragraph is then applied.

Example 3.7: *Rigidity constraints for a point on plane*
Suppose we have used a proximity sensor to observe a point of some planar face of an object. Let P be a reference associated to the location of the point, so that:

$$\begin{aligned} \hat{\mathbf{p}}_P &= 0 \\ C_P &= \text{diag}(\sigma_s^2, \sigma_s^2, \sigma_s^2) \end{aligned}$$

Let the estimated object location be $\mathbf{L}_{WO} = (\hat{\mathbf{x}}_{WO}, \hat{\mathbf{d}}_O, C_O)$, where:

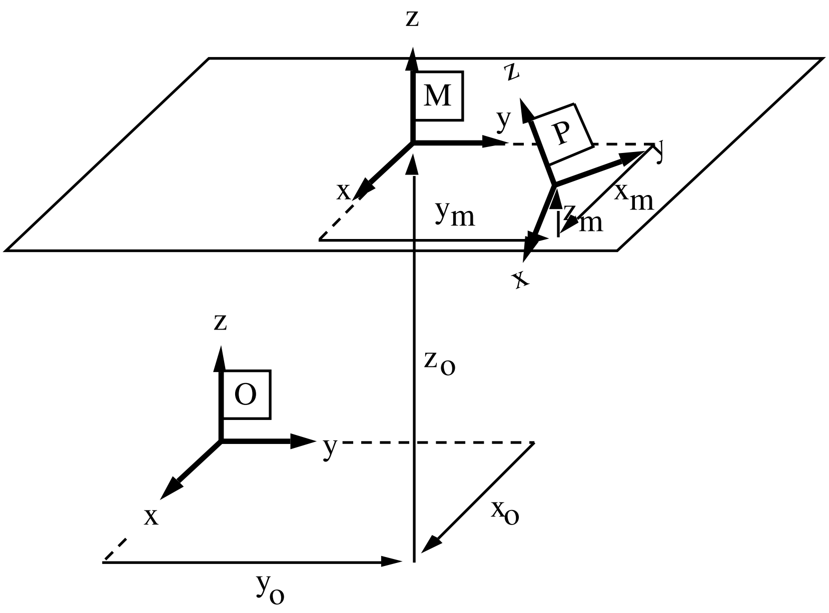


Figure 3.3: Validating the rigidity constraint for a point on a plane

$$
\begin{aligned}
\hat{\mathbf{d}}_O &= 0 \\
C_O &= \mathrm{diag}(\sigma_p^2, \sigma_p^2, \sigma_p^2, \sigma_o^2, \sigma_o^2, \sigma_o^2)
\end{aligned}
$$

The point has symmetries if rotations around any axis (R_{xyz}). Thus its self-binding matrix B_P is:

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

The plane has symmetries of translation along its x and y axes, and a symmetry of rotation around its z axis ($T_{xy}\,R_z$). Thus, its self-binding matrix B_M is:

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

The binding matrix of the pairing B_{MP} expresses the condition that the point belong to the plane, that is, the distance of the point to the plane in the z axis of the reference of the plane must be zero:

```
FUNCTION satisfy_rigidity_constraint ($\mathbf{L}_{WO}$, $p$)

; $\mathbf{L}_{WO}$: estimated object location
; $p = (e, m)$: proposed pairing
; $e = (\mathbf{L}_{WE}, \mathcal{S}_E)$ , $m = (\mathbf{x}_{OM}, \mathcal{S}_M)$

; determines whether the location of $e$ and $m$ can be considered compatible

    $\mathbf{L}_{WM} := \mathbf{L}_{WO} \oplus \mathbf{x}_{OM}$;
    $\hat{\mathbf{x}}_{EM} := \ominus\hat{\mathbf{x}}_{WE} \oplus \hat{\mathbf{x}}_{WM}$;
    $C_{EM} := J_{1\oplus}\{0, \hat{\mathbf{x}}_{EM}\} B_E^T C_E B_E J_{1\oplus}^T\{0, \hat{\mathbf{x}}_{EM}\} + J_{2\oplus}\{\hat{\mathbf{x}}_{EM}, 0\} B_M^T C_M B_M J_{2\oplus}^T\{\hat{\mathbf{x}}_{EM}, 0\}$;
    $D := (B_E\hat{\mathbf{x}}_{EM})^T \left[B_E C_{EM} B_E^T\right]^{-1} B_E\hat{\mathbf{x}}_{EM}$;
    $D_m :=$ chi_square (rank($B_E$), $\alpha$);

RETURN $D \leq D_m$;
END;
```

Algorithm 3.2: *Validating the Rigidity Constraint*

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

Let us consider a simple case, where $\mathbf{x}_{OM} = (x_o, y_o, z_o, 0, 0, 0)^T$ (fig. 3.3). According to (3.20), we have that the location of the model feature is given by $\mathbf{L}_{WM} = (\hat{\mathbf{x}}_{WM}, \hat{\mathbf{p}}_M, C_M)$, where:

$$C_M = \begin{bmatrix} \sigma_p^2 + (x_o^2 + y_o^2)\,\sigma_o^2 & y_o\,\sigma_o^2 & -x_o\,\sigma_o^2 \\ y_o\,\sigma_o^2 & \sigma_o^2 & 0 \\ -x_o\,\sigma_o^2 & 0 & \sigma_o^2 \end{bmatrix}$$

Suppose $\hat{\mathbf{x}}_{MP} = (x_m, y_m, z_m, \psi_m, \theta_m, \phi_m)^T$. Using (3.21) we can calculate $Cov(\mathbf{x}_{MP})$ and use it to calculate D^2, which yields:

$$D^2 = \frac{z_m^2}{\sigma_s^2 + \sigma_p^2 + ((x_m + x_o)^2 + (y_m + y_o)^2)\,\sigma_o^2}$$

The value of D^2 increases with z_m because it measures the discrepancy along the z axis, the only direction that is involved. D^2 decreases with the square of $x_m + x_o$ and $y_m + y_o$, making the test less discriminant for features whose location relative to the object is more distant. D^2 also decreases with the covariance of the errors of both the object location and the subfeature location, confirming that more precise features and object-location hypotheses will yield more discriminant constraint analysis. ◇

The implementation of `satisfy_rigidity_constraint` using the method given here is shown in algorithm 3.2.

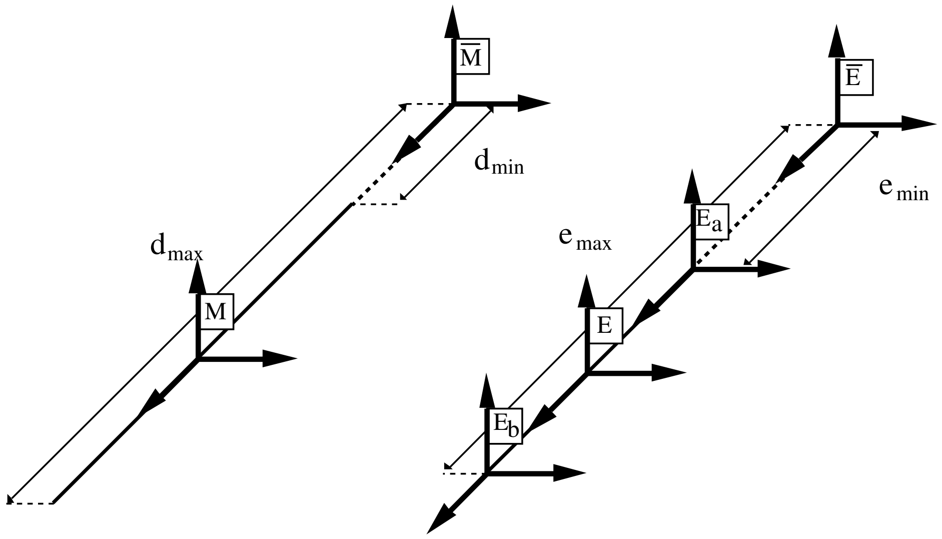


Figure 3.4: Applying the extension constraints to edges as a location independent constraint

3.3.2 The Extension Constraint

From the preceding example we can see that the rigidity constraint validates whether the point belongs to the *infinite* plane where its corresponding surface lies. It does not validate whether the point is actually located within the region occupied by the feature. This is also true for location independent constraints, since they correspond to parameters that are *invariant* under occlusion.

The validation that the observed feature is actually contained in the region occupied by the corresponding model feature is denominated the *extension* constraint. Unlike the other constraint validation methods, the extension constraint is particular for each type of geometric element. Given the estimated location of the model feature, it is very simple to validate it (determining whether the observed feature is located within the predicted region occupied by the model feature). Nevertheless, it can also be validated without having an estimation of the object location, that is, during the initial steps of the hypothesis generation process. In the following we give an example.

Example 3.8: *Validating the extension constraint for an edge*
Validating the extension constraint for an edge consists in determining whether the

most extreme points of the observed edge are contained within the extension occupied by its corresponding model edge. This validation is done after applying location independent constraints, using the aligned references obtained from this calculation (fig. 3.4). Let $\mathbf{L}_{WE_a} = (\hat{\mathbf{x}}_{WE_a}, \hat{\mathbf{p}}_{E_a}, C_{E_a})$ and $\mathbf{L}_{WE_b} = (\hat{\mathbf{x}}_{WE_b}, \hat{\mathbf{p}}_{E_b}, C_{E_b})$ represent the uncertain location of these points.

The procedure consists in estimating the distance between points E_a and E_b and the aligning reference $\bar{E}$. This can be done by obtaining an estimation of their relative location vector, and extracting its x component and its variance, which will correspond to $\hat{e}_{min}$, $Cov(e_{min})$ and $\hat{e}_{max}$, $Cov(e_{max.})$.

Given d_{min} and d_{max}, the extension constraint is satisfied if $\hat{e}_{min} \geq d_{min} \wedge \hat{e}_{max} \leq d_{max}$. If either of these conditions is not satisfied, then we must statistically test for equality. For example, if $\hat{e}_{min} < d_{min}$, we apply a hypothesis test on:

$$D^2 = \frac{(d_{min} - \hat{e}_{min})^2}{Cov(e_{min})}$$

For a significance level α we will consider d_{min} and e_{min} compatible if:

$$D^2 < D^2_{1,\alpha}$$

where $D^2_{1,\alpha}$ is the value of the χ^2_1 distribution for α. ◇

3.4 Conclusions

In this chapter we have shown how to derive the geometric relations that depend on the relative location and symmetries of the involved geometric elements, and how they can be expressed in the form of an uncertain location vector. This allows to define a general constraint validation mechanism, which can be applied to both location dependent and location independent constraints. This mechanism is based on statistical tests on the components of the location vector.

This constraint validation mechanism allows the use of diverse geometric information for recognition. Nevertheless, the discriminancy of constraint validation largely depends on the precision and relative location of the involved geometric features. That is, imprecise observations, or observations whose relative location is close to a particular case, have a greater probability of being spuriously paired. This highlights the importance of having criteria for the selection of the observations that will allow more discriminant constraint analysis. The fundamental concepts that allow to establish such criteria, the *precision* and *relevance* of an observation and of a hypothesis, are discussed in the next chapter. Criteria for the selection of observations and hypotheses are discussed in chapter 6.

The constraint validation mechanism proposed here can be used in the *identifying before locating*, as well as the *identifying while locating* approaches. However, since location independent constraints validate local consistency, while location dependent constraints validate global consistency, constraint validation in the *identifying while locating* scheme can be more discriminant and computationally less expensive. This subject is analyzed in chapter 6.

Chapter 4

Precision and Relevance

Summary

The fundamental goal of recognition is to obtain precise *and* relevant *object-location hypotheses. The precision of a hypothesis is related to the uncertainty in the estimation of the location of the hypothesized object. The relevance of a hypothesis is related to how confident the system may be in the assertion that the set of observations contained in the hypothesis comes from an instance of the hypothesized object. In this chapter we define two measurements used to characterize the precision and relevance of an object-location hypothesis [Neira 93c]. The proposed precision measurement allows the system to direct its attention to the observations and model features that can make the estimation more precise. The relevance measurement is general for any observation, and takes into account the precision of the sensor that obtained it, and the characteristics of the involved geometric feature. Both measurements allow us propose a set of recognition strategies that can speed up the recognition process (chapter 6).*

4.1 Introduction

Robotic object recognition systems have a twofold goal: the *identification* and *localization* of the objects present in the robot workspace. Identification can be considered a *search* problem, in which the goal is to find the best interpretation—in terms of the object models— of a given set of sensorial observations. Localization is an *estimation* problem, whose goal is to compute the object location where the discrepancy between the location of the observations and their corresponding model features is minimal.

One of the fundamental problems of geometric recognition is the characterization of degree of identification and localization of an object location hypothesis. The *precision* of a hypothesis is related to the uncertainty in the estimation of the location of the hypothesized object. The *relevance* of a hypothesis is related to how confident the system may be in the assertion that the set of observations contained in the hypothesis comes from an instance of the hypothesized object.

The characterization of the precision of an object-location hypothesis is closely related to the method used to model location uncertainty. In systems that represent location uncertainty using *set-based* models [Brooks 82, Brooks 85, Grimson 84, Grimson 90a], the precision of an observation or a hypothesis may be measured using a *bound* on the error of position and orientation. Apparently, this precision measurement has the advantage of simplicity, but computing the error bounds for a hypothesis given the error bounds of a set of observation can be very complex [Ellis 91, Fisher 91]. Additionally, such a precision measurement does not allow to determine whether there is more uncertainty in some directions than in others, and it tends to be conservative [Brooks 85, Smith 88]. Furthermore, *coupling* between position and orientation errors is seldom considered.

In probabilistic models, the precision of the estimation of a location is related to the eigenvalues of the covariance matrix of the estimation. The greater its eigenvalues, the more uncertain the estimation. Nakamura [Nakamura 89] uses the *volume* of uncertainty, computed as the product of the eigenvalues of the covariance matrix, as a measurement of the precision of an estimation. However, the volume of uncertainty does not give an idea of how much uncertainty there is on *each* main direction, and how uncertainty in position compares to uncertainty in orientation. An additional limitation of this approach is that during the initial steps of recognition the covariance matrix of the estimation may not be well-defined.

In section 4.2 we study this problem, and propose a precision measurement that overcomes these limitations. It is based on the *information matrix* of the object location estimation, the inverse of the covariance matrix.

The characterization of the *relevance* of a pairing between a model feature and an observation to the identification of an object, i.e., how much information the pairing contributes to support the object location hypothesis, has given rise to a great variety of solutions.

In the first place, some researchers have considered *heuristic approaches.* Faugeras [Faugeras 86] does not define a measure of the relevance of each feature. Instead, he applies three rules to determine the order in which the primitives are to be considered: avoid small primitives, choose linearly independent primitives—since they can be used to locate the object—, and choose primitives that discriminate between alternative positions of the object. In some systems, the relevance of a feature is heuristically preassigned in the object model, such as in [Allen 87], where the system assigns high priorities to model features which are large or isolated in space and protrude. This allows to order the features by their likelihood of being sensed. In [Granjean 91], a weight is assigned to each model feature, which is heuristically calculated from its discriminating power, extraction costs, reliability of extraction, and power of location. In [Lacroix 92], aspect graphs to recognize objects are built and ordered according to the rareness and robustness of the features composing the aspect.

Bolles and Cain [Bolles 82] propose a method to automatically select *focus features* of the CAD models of 2D objects for their recognition and localization using vision. The selection of these feature depends on their uniqueness, expected contribution, cost of detection and likelihood of detection. The system concentrates in finding one focal feature in the image, and uses it to predict nearby features to look for. This approach has been extended to the three dimensional case in [Bolles 86].

Different *formalization efforts* have also been done. In [Hutchinson 89], Hutchinson uses a Dempster-Shafer formalism to express the credibility of a hypothesis as belief functions of the set of observations, combined using Dempster's rule. Bhandarkar [Bhandarkar 92] uses fuzzy sets to describe qualitative attributes of object features, such as degree of occlusion and degree of satisfaction of geometric constraints to calculate a weight to each match and estimate the object location using a Weighted Generalized Hough Transform. Nagata [Nagata 91] also uses fuzzy sets to determine the degree of visibility of a model surface, to select the next one that the system should verify. In [Chen 90], Chen defines three feature utility measurements: detectability, reliability and error rate, and uses them to reduce the execution time of a vision program.

In [Grimson 90b], Grimson and Huttenlocher relate the relevance of a pairing to the probability that it be the result of a *conspiracy.* That is, the probability that an observation from a feature other than the one to which is being paired is located within its *uncertainty volume.* This volume depends on bounds on the degree of sensor noise. Grimson and Huttenlocher derive a threshold for the fraction of model features that must be paired so that the probability of random conspiracy is small. In [Tardós 91], Tardós proposes a similar approach, in which the probability of a *random* conspiracy is related to the *volume of acceptance* of the observation, and the uniqueness of the feature in the models. Tardós points out the limitations of this approach, which does not take into account the probability of an *organized conspiracy,* related to the similarities between the object models.

The fundamental limitation of most of these approaches is that they concentrate on specific types of features, for example edges. In multisensor systems, where different sensors may be able to obtain information for a great variety of geometric features, it is not possible to compare the potential benefit of selecting observations corresponding to different types of geometric features, say edges, planes, corners or dihedrals. A second limitation that one sole bound for sensor error is considered. The use of several sensors requires taking into account their precision and reliability.

In section 4.3 we propose an approach to characterize the relevance of a pairing and of a hypothesis, which is closely related to the mechanism for integrating uncertain geometric information, and thus has all the advantages of generality, and feature and sensor sensibility that characterize such integration mechanism. The characterization of relevance of a pairing will be extended to determine the relevance of a sensorial observation, and the potential relevance of a model feature.

4.2 The Precision of an Object-Location Hypothesis

4.2.1 Regions of Uncertainty

Consider a geometric element e, whose uncertain location with respect to a reference W is expressed by $\mathbf{L}_E = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E, C_E)$. Intuitively, the precision of the estimated location of the geometric element is related to how much its true value can deviate from its estimated value, that is, to its covariance. We can measure how much the true value of the perturbation vector $\mathbf{p}_E$ deviates from $\hat{\mathbf{p}}_E$ using the *Mahalanobis distance* [Cuadras 89]:

$$D^2 \triangleq (\mathbf{p}_E - \hat{\mathbf{p}}_E)^T \; C_E^{-1} \; (\mathbf{p}_E - \hat{\mathbf{p}}_E)$$

Given that $\mathbf{p}_E \sim N(\hat{\mathbf{p}}_E, C_E)$, we have that D^2 follows a chi-square distribution χ^2_m, where $m = dim(\mathbf{p}_E)$ is the number of d.o.f. that define the location of the geometric element. A significance level α defines an m-dimensional region where $\mathbf{p}_E$ will lie with probability $1-\alpha$:

$$P\left\{D^2 \leq D^2_{m,\alpha}\right\} = 1-\alpha$$

where $D_{m,\alpha}$ is the value of χ^2_m for a significance level α. Given m and α, this value can be obtained from the chi-square tables [Ríos 77]. We call such region an *uncertainty region* (with probability $1-\alpha$). To graphically represent regions of uncertainty, we will assume a significance level of $\alpha = 0.05$, that is, we will draw regions corresponding to a probability of 0.95.

The region defined by the condition $D^2 \leq D^2_{\alpha,m}$ is limited by an ellipsoid whose equation is:

$$(\mathbf{p}_E - \hat{\mathbf{p}}_E)^T \; C_E^{-1} \; (\mathbf{p}_E - \hat{\mathbf{p}}_E) = D^2_{m,\alpha} \tag{4.1}$$

The principal axes of this ellipsoid have length $D_{m,\alpha}\sqrt{\lambda_i}$, where λ_i are the eigenvalues of C_E. Let us illustrate this fact using a two dimensional example.

Example 4.1: *The location of a vertex in 2D*
Consider that the geometric element e corresponds to a vertex in two dimensional space. In this case we have $m = 2$ and:

$$\mathbf{p}_E = (d_x,\ d_y)^T \quad ; \quad C_E = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}$$

Using (4.1), we can calculate the equation of the ellipse that contains the region of uncertainty with probability $1-\alpha$ (figure 4.1), which gives:

$$\frac{\sigma_y^2\, d_x^2 - 2\; d_x\, d_y\, \sigma_{xy} + \sigma_x^2\, d_y^2}{\sigma_x^2\, \sigma_y^2 - \sigma_{xy}^2} = D^2_{2,\alpha}$$

Considering that $\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x\,\sigma_y}$, the equation becomes:

$$\frac{d_x^2}{\sigma_x^2} + \frac{d_y^2}{\sigma_y^2} - \frac{\rho_{xy}\, d_x\, d_y}{\sigma_x\, \sigma_y} = D^2_{2,\alpha}\,(1-\rho_{xy}^2)$$

The value of $D^2_{2,\alpha}$ can be obtained from the chi-square tables [Ríos 77]. For example, for $\alpha = 0.05$, we have that $D_{2,0.05} = 2.44$.

The eigenvalues of the covariance matrix C_E are:

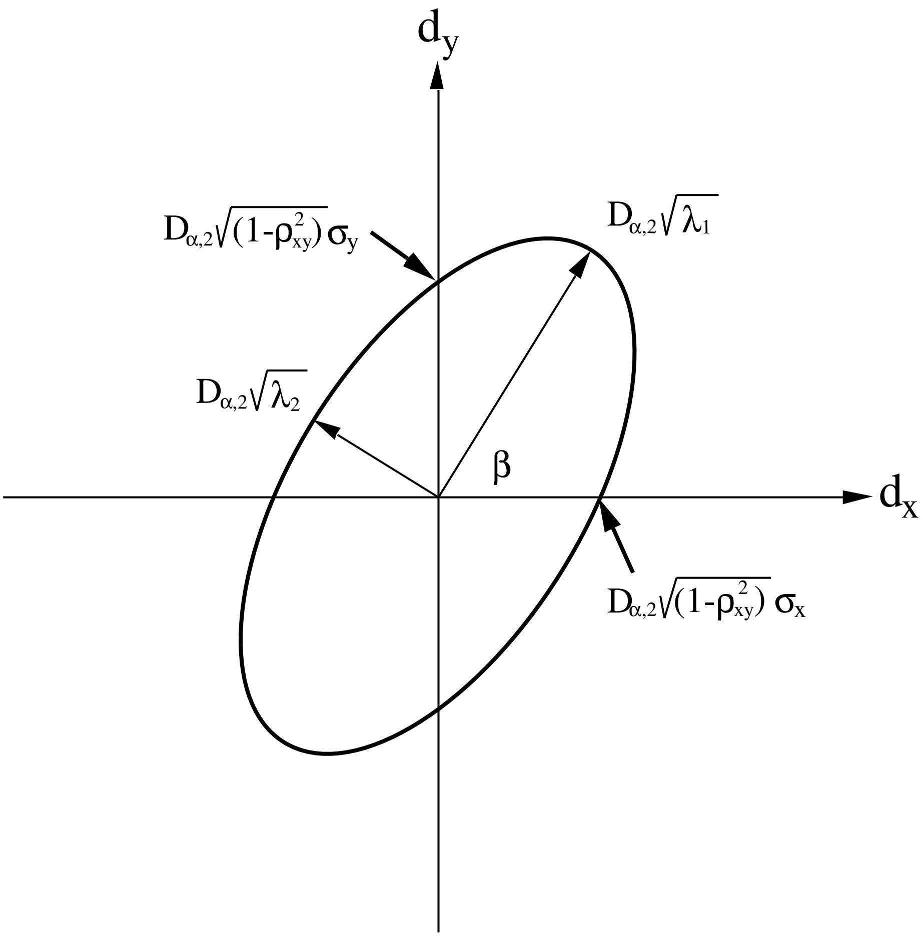

d_y
$D_{\alpha,2}\sqrt{(1-\rho_{xy}^2)}\,\sigma_y$
$D_{\alpha,2}\sqrt{\lambda_1}$
$D_{\alpha,2}\sqrt{\lambda_2}$
β
d_x
$D_{\alpha,2}\sqrt{(1-\rho_{xy}^2)}\,\sigma_x$

Figure 4.1: Ellipse of uncertainty of a 2D vertex

$$\begin{aligned}
\lambda_1 &= \frac{\sigma_x^2 + \sigma_y^2 + \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4\,\sigma_{xy}^2}}{2} \\
\lambda_2 &= \frac{\sigma_x^2 + \sigma_y^2 - \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4\,\sigma_{xy}^2}}{2}
\end{aligned}$$

The eigenvectors corresponding to these eigenvalues give the directions along which the value of $\mathbf{p}_E$ can drift more or less from its mean value. That is, the directions of highest and lowest uncertainty, respectively ($\lambda_1 \geq \lambda_2$). The angle β between the eigenvector related to the largest eigenvalue and the x axis is given by:

$$\beta = \frac{1}{2}\tan^{-1}\left(\frac{2\,\sigma_{xy}}{\sigma_x^2 - \sigma_y^2}\right)$$

If $\sigma_{xy} = 0$, then $\beta = 0$ or $\beta = \pi/2$ and the eigenvectors coincide with the axes. When $\sigma_x^2 = \sigma_y^2$, we have that $\beta = \pm\pi/4$. If these two conditions are simultaneous, then the value of β is indeterminate because we are in the degenerate case when the ellipse becomes a circle. In this case, $\lambda_1 = \lambda_2 = \sigma_x = \sigma_y$ and all vectors are eigenvectors.

In this two dimensional case the area of the ellipse is given by:

$$\begin{aligned}
A &= \pi\, D_{2,\alpha}^2\, \lambda_1\, \lambda_2 \\
&= \pi\, D_{2,\alpha}^2\, \det(C_E)
\end{aligned}$$

◇

This example shows that the precision of an observation is fundamentally related to the eigenvalues of the covariance matrix. The smaller these eigenvalues are, the smaller the volume of uncertainty becomes.

Let us now generalize the computation of the volume of uncertainty to the estimated location of an object. Consider an object whose uncertain location with respect to W is given by $\mathbf{L}_{WO} = (\hat{\mathbf{x}}_{WO}, \hat{\mathbf{d}}_O, C_O)$. For a given significance level α, the volume of the ellipsoid defined by the condition $\mathbf{d}_O^T\, C_O^{-1}\, \mathbf{d}_O < D_{\alpha,m}^2$ is given by [Tardós 91]:

$$\begin{aligned}
V &= k_m\, D_{m,\alpha}^m \prod_{i=1}^{m} \lambda_i \\
&= k_m\, D_{m,\alpha}^m\, \det(C_O)
\end{aligned}$$

where k_m is a constant which depends of the number of d.o.f. that define the location of a geometric element ($k_1 = 2$, $k_2 = \pi$, $k_3 = 4\pi/3$, $k_4 = \pi^2/2$, $k_5 = 8\pi^2/15$, $k_6 = \pi^3/6$), and $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n$ are the eigenvalues of C_O.

An important property of this volume of uncertainty is that it remains constant under change of reference. If we change the reference of the object O to O', the covariance of the perturbation vector in O' will be:

$$C_{O'} = J_{O'O} \, C_O \, J_{O'O}^T$$

where $J_{O'O}$ is the Jacobian of the transformation between O' and O (see appendix A). The determinant of $C_{O'}$ will be:

$$\begin{aligned} \det(C_{O'}) &= \det(J_{O'O} \, C_O \, J_{O'O}^T) \\ &= \det(J_{O'O}) \det(C_O) \det(J_{O'O}^T) \end{aligned}$$

Given that the determinant of the Jacobian of any transformation is equal to 1 (appendix A), we have $\det(J_{O'O}) = 1$ and thus the volume of uncertainty remains constant:

$$\det(C_{O'}) = \det(C_O)$$

The relation between the volume of uncertainty and the eigenvalues of the covariance matrix will help us in defining a measurement for the precision of an object location hypothesis, but before we can go any further, we must study the relation between position and orientation errors.

4.2.2 The Influence of Orientation Errors in Position Errors

The location of an object is determined by three position and three orientation parameters. Since the covariance matrix contains information related to the uncertainty of both components of the location vector, the eigenvalues of the covariance matrix mix position and orientation terms. Therefore, we must find a way of expressing all the elements in the covariance matrix in the same units.

An approach to solving this problem lies in the fact that the uncertainty in the orientation of an object affects the uncertainty in the position of points of the object. Consider the example in figure 4.2, where the location of two vertices is predicted from the estimated location of the object. (For ease of visualization, ellipses around the point are used to depict position errors, while cones around an axis are used to depict orientation errors; in some degenerate cases, ellipses become lines as in fig. 4.2.c.) If the object location is only affected by uncertainty in position (a), then all points of the object are affected by the same position error. If the object location is

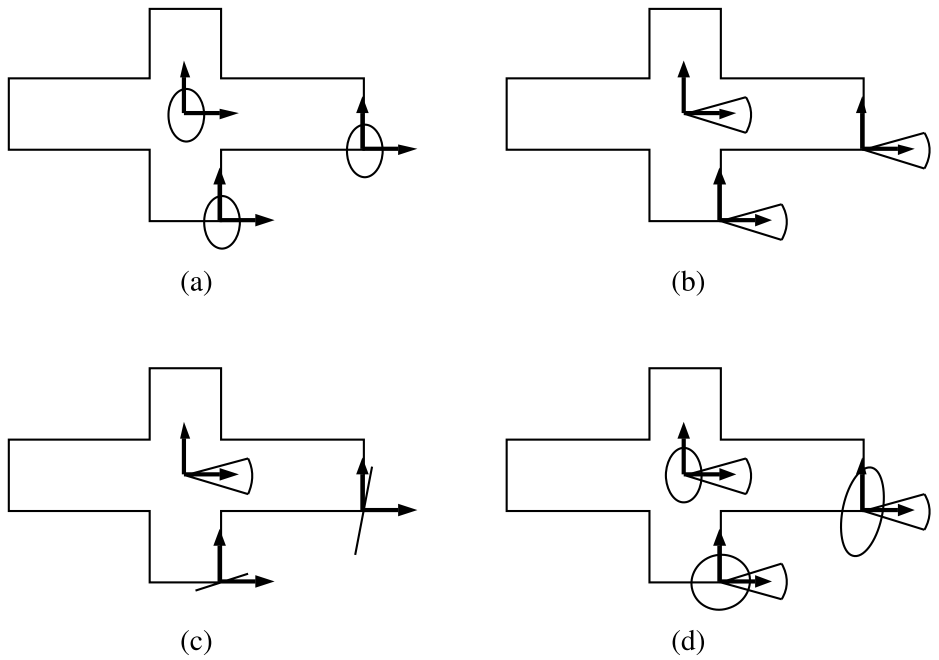


Figure 4.2: Object location errors affect feature location errors

affected by an orientation error, this has an effect in the orientation of the points (b), and also in their position errors (c). The combined effect of object location position and orientation errors in the location of features is shown in (d). In general, the position error of a feature due to the orientation error of the object is proportional to the distance between the feature reference and the object reference. This gives us three clues in managing this problem:

Choosing an adequate object reference: In order to reduce the effect that the orientation error of the object has in the position error of its features, the object reference should be located near the geometric center of the object, where the mean square distance to all surface points is minimal (fig. 4.3).

Normalize the covariance matrix: Orientation errors can be weighted by their effect on position errors. This results in *normalizing* the covari-

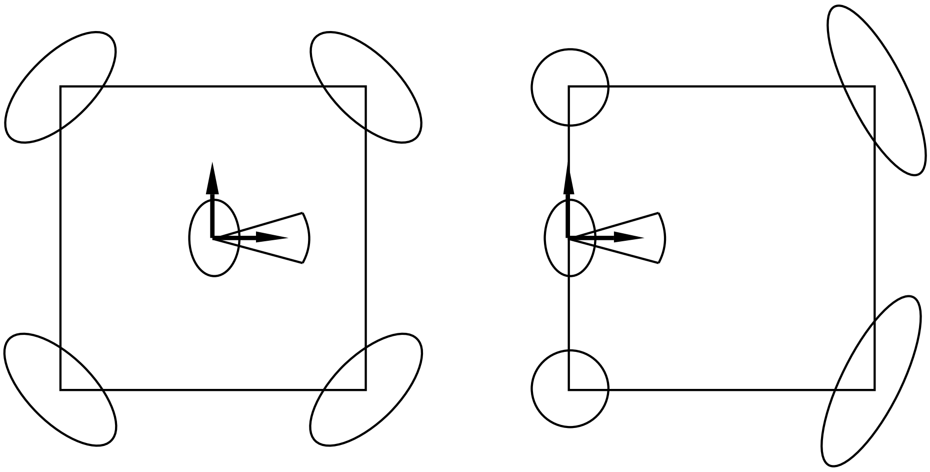


Figure 4.3: Choosing a reference near the center of the object reduces the coupling effect

ance matrix so that is has only units of position. The normalization of the covariance matrix of any geometric element can be done in the following way: let p and o by the number of position and orientation terms of the perturbation vector $\mathbf{p}_E$ of the uncertain location of the geometric element. The normalization is done by means of a matrix N_E of the form:

$$N_E = \mathrm{diag}(\underbrace{1, \cdots, 1}_{p}, \underbrace{r, \cdots, r}_{o})$$

where r is a weighting factor for orientation errors. The normalized covariance matrix will be:

$$\overline{C}_E = N_E \, C_E \, N_E^T \tag{4.2}$$

In this way we obtain a covariance matrix that expresses all location errors in the same units. The normalized volume of uncertainty is easily derivable from (4.2):

$$\begin{aligned} \det(\overline{C}_E) &= \det\left(N_E \, C_E \, N_E^T\right) \\ &= \det(N_E) \det(C_E) \det(N_E^T) \\ &= r^o \det(C_E) \, r^o \end{aligned}$$

$$= \quad r^{2\,o}\det(C_E)$$

Select an appropriate value for r: The value of r can be adjusted to magnify or reduce the importance of orientation errors. The value of r should be related to the object or feature size. We give r the value of the mean radius of the object or feature.

Example 4.2: *Edges observed with 2D vision*
Suppose we use 2D vision to obtain two images of an edge to determine its estimated location in the scene, as described in chapter 2. If these two images are obtained from two orthogonal directions, from (2.19) we have:

$$C_E = \frac{s\,\sigma_p^2}{l}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{12}{l^2} & 0 \\ 0 & 0 & 0 & \frac{12}{l^2} \end{bmatrix}$$

where l is the edge length and s the scale factor related to the camera focus and its distance to the edge (in mm/pixel). In this case the form of N_E is:

$$N_E = \operatorname{diag}(1, 1, r, r)$$

The mean distance between points of an edge of length l and its center is $l/4$. Giving this value to r, from (4.2) we would obtain the following normalized covariance matrix:

$$\overline{C}_E = \frac{s\,\sigma_p^2}{l}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3/4 & 0 \\ 0 & 0 & 0 & 3/4 \end{bmatrix}$$

We can see that we obtain a covariance matrix with all its components expressed in the same units. The results show that, in average, the orientation error of the edge will produce a position error $\sqrt{3/4}$ times the error produced by the position error of the edge. ◇

4.2.3 A Definition and some Properties of Precision

Let h be an object location hypothesis, composed of n pairings between model features and sensorial observations. Let Q_n be the information matrix of the estimation of the object location after integrating the n measurements. Matrix Q_n is the result of adding the information matrices of each of the n pairings, as expressed in (2.7):

$$Q_n = \sum_{k=1}^{n} F_k$$

Thus, this matrix is well defined during all the steps of the integration process. Let $\mathbf{L}_{WO} = (\hat{\mathbf{x}}_{WO}, \hat{\mathbf{d}}_O, C_O)$ represent the uncertain location of the object, obtained by integrating the n pairings. We have that:

$$C_0 = Q_n^{-1}$$

Let $\overline{Q}_n$ be the normalized information matrix, where:

$$\overline{Q}_n = N_O^{-T}\, Q_n\, N_O^{-1}$$

So far we have seen that we can relate the precision of the estimated object location with the probability that $\mathbf{d}_O$ deviates from $\hat{\mathbf{d}}_0$ more than a given limit. This limit defines an m-dimensional ellipsoid. We have also seen that the main directions of uncertainty are given by the eigenvectors of the covariance matrix C_O which constitute the main axes of the ellipsoid. In this way, their corresponding eigenvalues characterize the precision of the estimation. We have seen that the determinant of the covariance matrix C_O is related to the product of the eigenvalues, and that this volume remains invariant under change of reference. These reasons seem to suggest the use of the volume of uncertainty as the precision measurement [Nakamura 89]. Nevertheless, it has some fundamental drawbacks:

- Since the volume is the product of the eigenvalues of the covariance matrix, it may occur that there is considerable uncertainty in one direction, and very little in some other, and they would compensate in the product.

- It is not well defined in all the integration steps, especially in the initial ones, when the set of observations may not be sufficient to completely determine the object location. In this case, the information matrix Q_n is singular, and thus, the covariance matrix contains infinite terms.

These limitations can be overcome by using an approximation slightly different from that of the volume of uncertainty. This approximation follows two ideas:

- Use the inverse of the covariance matrix, the information matrix. Since the inverse of the eigenvalues of the covariance matrix are the eigenvalues of the information matrix, they provide equivalent ways of defining precision, with the advantage that the information matrix is well defined throughout all the steps of the integration process.

- Consider only the amount of uncertainty in the most uncertain direction. This direction is related to the smallest eigenvalue of the information matrix (e.g. the largest eigenvalue of the covariance matrix). A direction of high uncertainty means a large eigenvalue in the covariance matrix, or equivalently a small eigenvalue of the information matrix. The difference is that when uncertainty in this direction is infinite, the covariance matrix is not defined, while the eigenvalue of the information matrix becomes zero.

Definition 4.1: Precision of an Object Location Hypothesis
Given an object location hypothesis h, composed of n pairings between model features and sensorial observations; given $\overline{Q}_n$, the normalized information matrix of the estimation of the object location after integrating the n measurements; and given $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_6$, the eigenvalues of $\overline{Q}_n$, we define the precision of the estimated location of the object as:

$$\begin{aligned} \texttt{precision}(h) &\triangleq \sqrt{\min\left\{\text{eigenvalues}\left(\overline{Q}_n\right)\right\}} \\ &= \sqrt{\lambda_6} \end{aligned} \tag{4.3}$$

Next, we will describe the fundamental properties of the measurement of precision that we have defined.

1. *Non-negativity*
   The value of precision is a non-negative real number. This property derives from the fact that the information matrix is a positive, semidefinite matrix. Thus, all its eigenvalues are non-negative real numbers.

2. *Well-definedness*
   Precision is always well defined because the information matrix is well defined throughout all the steps of the integration process. Thus, we can always calculate its eigenvalues. In the case where the information matrix is singular (the covariance matrix cannot be calculated), at least one of the eigenvalues of the information matrix must be zero. Since all its eigenvalues are non-negative, the resulting precision is zero. This means that there is some direction in which uncertainty is infinite.

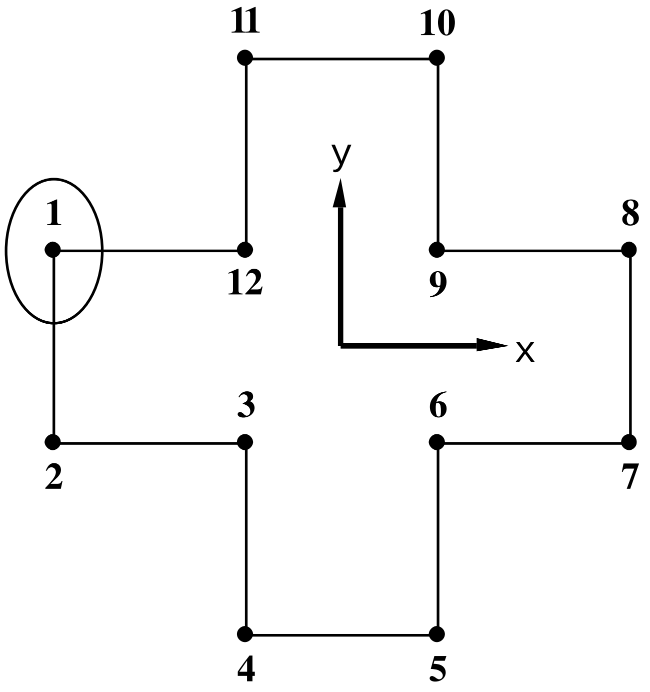


Figure 4.4: Object and observation models for simulation

3. *Non-decreasing*
The precision of an object location hypothesis is a non-decreasing function of the integrated observations. This derives from the fact that the smallest eigenvalue of the sum of two matrices is greater than or equal to the sum of the smallest eigenvalues of the involved matrices [Franklin 68]. This property guarantees that the integration of a pairing in the hypothesis will not result in a loss of precision in the estimation. In case of redundant observations, precision remains constant.

4. *Units of distance*$^{-1}$
If we describe position errors in millimeters, then all the elements of the normalized information matrix $\overline{Q}_n$ have units of 1/mm^2. This means that the units of the square roots of its eigenvalues are 1/mm.

4.2.4 An Example with Simulated Data

In order to analyze the evolution of the precision of an object location hypothesis we will study the case of the estimation of the location of the object shown in figure 4.4. We will assume that the length of each edge is 16 millimeters. Suppose we have a sensor that can detect vertices, and the precision of the sensor is such that an observed vertex will have covariance equal to $\mathrm{diag}(\sigma_x^2, \sigma_y^2)$, where $\sigma_x = 2mm$ and $\sigma_y = 3mm$.

Let us see how the precision of the hypothesis evolves when the observed vertices are integrated in the order in which they are numbered. The results of such an integration process are shown in figure 4.5.a. Given that two non-coincident vertices completely determine the location of an object in 2D space, the precision of the hypothesis becomes grater than zero only after integrating the second observation. We can also see that the first integrated observations have more impact on precision the the last ones. In general, the contribution of an observation to the precision in the estimated location of the object depends not only on the precision of the observation itself, but also on its relative location to the observations that have already been integrated in the hypothesis. This relative location conditions how much information the observation will contribute in each direction, and how it will contribute to reduce the coupling between position and orientation.

This can be best seen by considering integrating the observations in a different order. Figure 4.5.b shows the evolution of the precision of the hypothesis integrating the observations in this order: $\{1, 7, 4, 10, 5, 11, 8, 2, 9, 3, 12, 6\}$. This sequence emerges when considering at each step the observation that would most contribute to the precision of the estimation. In this case, the initial observations increment more the precision of the hypothesis than in the previous case because they are more distant. In conclusion, the selection of the observations has an important impact in the precision of the estimated object location. Strategies that can aid the system in performing this selection will be described in chapter 6.

4.3 The Relevance of Hypotheses and Observations

The fundamental characteristics that are required from a relevance measurement to be used in multisensor systems are:

- *Generality:* It is not desirable to tie the recognition scheme to any particular type of feature. Thus, it should be possible to compute and

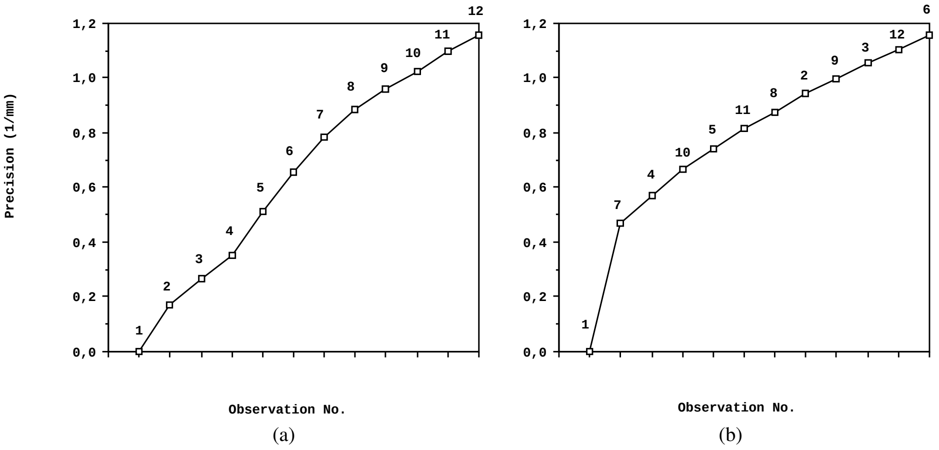


Figure 4.5: Evolution of precision for observations: (a) integrated in sequential order; (b) integrated in an alternative order

compare the relevance of all types of observations and features.

- *Feature Sensibility:* Nevertheless being general, the relevance of an observation must be a function of feature characteristics, such as its size, the number of degrees of freedom that determine its location, and the abundance of the feature in the models. Such characteristics are related to how many observations are needed to completely determine the object location and to how many alternate interpretations for an observation can be found in the models.

- *Sensor Sensibility:* The precision in the location of an observation depends on the sensor precision, and on the sensing strategy. This precision influences the probability of pairing it with an incorrect model feature, and the discriminancy of constraint analysis. Thus, it should be taken into account to compute relevance (additional aspects, such as reliability and sensing cost, which can be considered in sensor selection processes, are not considered here).

In the following we propose a relevance measurement that satisfies these requirements.

4.3.1 The Relevance of a Pairing

The requirements of a relevance measure that we have outlined in the introduction lead us to study the integration mechanism used in estimating the location of an object. It is a general integration mechanism, in which sensors and feature characteristics are reflected in the contribution of a pairing to the estimated object location. Two fundamental facts arise in the study of the integration mechanism:

- Sensor and feature characteristics are reflected in the covariance matrix of the observation.
- The amount of location information contributed by a pairing is contained in the information matrix of the pairing, and the location information related to a hypothesis is contained in the information matrix of the estimation.

Let us explore these ideas in further detail.

The Covariance Matrix of an Observation

Two fundamental feature characteristics are reflected in the covariance matrix of an observation: its size and the d.o.f. that define its location. In general, size is reflected in the covariance matrix because under equivalent sensing conditions, a sensor will be able to collect more information from larger features. The number of degrees of freedom that determine the location of the feature are reflected in the dimensions of its perturbation vector, and thus, its covariance matrix.

Example 4.3: *The covariance matrix of a proximity edge*
If we use a proximity sensor to obtain an observation of an edge, as explained in chapter 2, the covariance matrix of the resulting observation will be:

$$C_E = \operatorname{diag}\left(\frac{\sigma_y^2}{n}, \frac{\sigma_z^2}{n}, \frac{12\,\sigma_z^2}{n^3\,s^2}, \frac{12\,\sigma_y^2}{n^3\,s^2}\right)$$

where n is the number of observed points, s the distance between each observation, and σ_y and σ_z represent the precision of the sensor. If we consider that the edge has been completely observed, then $n\,s = l$, where l is the edge length. Thus, the size of the edge is represented by the values of n and s. This covariance matrix has dimension four: two for edge position and two for orientation. ◇

Sensor characteristics are also directly reflected in the covariance matrix of the observation (in the example the sensor is represented by σ_x and σ_y).

Furthermore, the sensing strategy used in obtaining the observation may also influence the resulting covariance matrix. In this case, the distance d between observed points will determine how many points will be collected.

The Information Matrix of a Pairing

Given a pairing (e_k, m_k), the information that the pairing contributes to the object location is contained in the information matrix of the pairing F_k. Suppose the uncertain location of e_k is represented by $\mathbf{L}_{WE_k}$, and the predicted location of model feature m_k is given by $\mathbf{L}_{WM_k}$. In chapter 2 we have seen that from (2.8) this information matrix F_k is calculated as :

$$F_k = H_k^T (G_k S_k G_k^T)^{-1} H_k$$

In the case of the integration of a feature to estimate the location of an object, from (2.11) we have:

$$\begin{array}{rcl} H_k & = & (B_{E_k} J_{2\oplus}\{\hat{\mathbf{x}}_{E_k M_k}, 0\})\, J_{M_k O} \\ G_k & = & -B_{E_k} J_{1\oplus}\{0, \hat{\mathbf{x}}_{E_k M_k},\} B_{E_k}^T \\ S_k & = & C_{E_k} \end{array}$$

Thus, F_k is calculated as follows:

$$\begin{array}{rcl} F_k & = & (B_{E_k} J_{2\oplus}\{\hat{\mathbf{x}}_{E_k M_k}, 0\}\, J_{M_k O})^T \\ & & (B_{E_k} J_{1\oplus}\{0, \hat{\mathbf{x}}_{E_k M_k},\} B_{E_k}^T C_{E_k} B_{E_k} J_{1\oplus}^T\{0, \hat{\mathbf{x}}_{E_k M_k},\} B_{E_k}^T)^{-1} \\ & & (B_{E_k} J_{2\oplus}\{\hat{\mathbf{x}}_{E_k M_k}, 0\}\, J_{M_k O}) \end{array} \tag{4.4}$$

There are four essential aspects of the relationship between the characteristics of the pairing to be integrated and its contribution to the location of the object that are captured in this expression:

- The symmetries of the involved geometric element, represented by B_E. Symmetries are related to the degrees of freedom that determine the location of the feature.

- The discrepancy between the predicted location of the model feature, and the estimated location of the observation, given by $\hat{\mathbf{x}}_{E_k M_k}$, which affects the value where we choose to linearize the measurement equation.

- The relative location of the feature with respect to the object, represented by J_{M_kO}.
- The quality of the observation, represented by its covariance matrix C_{E_k}.

In the case of the integration of a subfeature p_k, paired with a model feature m_k, from (2.12)we have :

$$\begin{aligned} H_k &= -B_{M_kP_k} J_{1\oplus}\{0, \hat{\mathbf{x}}_{M_kP_k}\} J_{M_kO} \\ G_k &= B_{M_kP_k} J_{2\oplus}\{\hat{\mathbf{x}}_{M_kP_k}, 0\} B_{P_k}^T \\ S_k &= C_{P_k} \end{aligned}$$

In this case F_k becomes:

$$\begin{aligned} F_k = & \left(B_{M_kP_k} J_{1\oplus}\{0, \hat{\mathbf{x}}_{M_kP_k}\}\, J_{M_kO}\right)^T \\ & \left(B_{M_kP_k} J_{2\oplus}\{\hat{\mathbf{x}}_{M_kP_k}, 0\} B_{P_k}^T C_{P_k} B_{P_k} J_{2\oplus}^T\{\hat{\mathbf{x}}_{P_kM_k}, 0\} B_{M_kP_k}^T\right)^{-1} \\ & \left(B_{M_kP_k} J_{1\oplus}\{0, \hat{\mathbf{x}}_{M_kP_k}\}\, J_{M_kO}\right) \end{aligned} \tag{4.5}$$

Similar factors relate this information matrix to the characteristics of the involved geometric feature and to the sensor that obtains the observation. An additional element in this case is the binding matrix of the pairing $B_{P_kM_k}$, which determines which d.o.f. of the subfeature location contribute information to the object location.

The general form of the information matrix of a pairing is:

$$F_k = \begin{bmatrix} T_k & U_k \\ U_k^T & R_k \end{bmatrix}$$

where T_k represents the contribution of the pairing to the estimation of the object position, R_k is the contribution to orientation, and U_k represents the coupling between the position and orientation terms. In the preceding section we have seen that the location information contained in the estimation of an object location can be characterized by the eigenvalues of its information matrix Q_n. Since the trace of a matrix is equal to the sum of its eigenvalues [Franklin 68], and trace is a linear operator, the contribution of the pairing to the position and orientation of the object can be characterized by the trace of the T_k and R_k submatrices. Matrix U_k contains information related to how a change in the value of the position parameters would influence the orientation parameters, and viceversa. This coupling effect has influence in the eigenvalues of the information matrix, but they have no influence on their *sum*.

Example 4.4: *The information matrix of a vertex pairing*
Suppose we have an sensor observation of a vertex, whose covariance matrix is:

$$C_{E_k} = \operatorname{diag}\left(\sigma^2, \sigma^2, \sigma^2\right)$$

The binding matrix of this observation is:

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

Suppose we pair it with a model vertex m_k, whose relative location with respect to the object location is given by:

$$\mathbf{x}_{OM_k} = (x,\ y,\ z,\ 0,\ 0,\ 0)^T$$

The Jacobian of $\mathbf{x}_{M_kO}$ will be:

$$J_{M_kO} = \begin{bmatrix} 1 & 0 & 0 & 0 & z & -y \\ 0 & 1 & 0 & -z & 0 & x \\ 0 & 0 & 1 & y & -x & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Suppose the relative location between the observation and the model feature is given by:

$$\hat{\mathbf{x}}_{E_kM_k} = (x_d,\ y_d,\ z_d,\ \psi_d,\ \theta_d,\ \phi_d)^T$$

Thus, according to (4.4), the information matrix of the pairing would be:

$$F_k = \frac{1}{\sigma^2} \begin{bmatrix} 1 & 0 & 0 & 0 & z & -y \\ 0 & 1 & 0 & -z & 0 & x \\ 0 & 0 & 1 & y & -x & 0 \\ 0 & -z & y & z^2+y^2 & -yx & -zx \\ z & 0 & -x & -yx & x^2+z^2 & -zy \\ -y & x & 0 & -zx & -zy & y^2+x^2 \end{bmatrix}$$

where x, y, and z represent the relative location between the object and the vertex. In this case, from the diagonal elements of the F_k matrix we can see that the pairing contributes $\frac{1}{\sigma^2}$ to the position of the object on each axis. The trace of the position submatrix T_k is:

$$\text{trace}(T_k) = \frac{3}{\sigma^2}$$

This means that the vertex contributes position information in three d.o.f., with the precision given by the sensor.

The contribution of the pairing to the orientation of the object around each axis is equal to the square distance of the vertex to the corresponding axis. The trace of the orientation submatrix R_k is:

$$\text{trace}(R_k) = 2\,\frac{x^2 + y^2 + z^2}{\sigma^2}$$

This means that the total contribution to orientation is equal to twice the square distance between the vertex and the object reference. ◇

Note that the contribution of the pairing to the position of the object is independent of the relative location of the vertex with respect to the object, while the contribution to orientation depends on the relative location (more distant vertices contribute more information). This is true for any observation.

Example 4.5: *The information matrix of a proximity edge pairing*
Consider a pairing including an edge observed using proximity, whose covariance matrix is (chapter 2):

$$C_{E_k} = diag\left(\frac{\sigma^2}{n}, \frac{\sigma^2}{n}, \frac{12\,\sigma^2}{n^3\,s^2}, \frac{12\,\sigma^2}{n^3\,s^2}\right)$$

The binding matrix of an edge is:

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

Suppose this observation is paired with a model edge m_k, whose location relative to the object location is given by:

$$\mathbf{x}_{OM_k} = (x,\ y,\ z,\ \psi,\ \theta,\ \phi)^T$$

The Jacobian of the location vector $\mathbf{x}_{M_kO}$ is:

$$J_{M_kO} = \begin{bmatrix} n_x & n_y & n_z & pn_x & pn_y & pn_z \\ o_x & o_y & o_z & po_x & po_y & po_z \\ a_x & a_y & a_z & pa_x & pa_y & pa_z \\ 0 & 0 & 0 & n_x & n_y & n_z \\ 0 & 0 & 0 & o_x & o_y & o_z \\ 0 & 0 & 0 & a_x & a_y & a_z \end{bmatrix}$$

where n_x, n_y, n_z, o_x, o_y, o_z, a_x, a_y, a_z, pn_x, pn_y, pn_z, po_x, po_y, po_z, pa_x, pa_y, and pa_z are the values of the homogeneous matrix representing the relative location between the object and the edge (see appendix A).

For simplicity, we will consider that the predicted location of the model feature and the location of the observed edge coincide, that is:

$$\hat{\mathbf{x}}_{E_k M_k} = (0,\ 0,\ 0,\ 0,\ 0,\ 0)^T$$

Computing the value of the F_k matrix from (4.4) we have that the corresponding submatrix of position T_k is:

$$T_k = \frac{n}{\sigma^2} \begin{bmatrix} o_x^2 + a_x^2 & o_x\, o_y + a_x\, a_y & o_x\, o_z + a_x\, a_z \\ o_x\, o_y + a_x\, a_y & o_y^2 + a_y^2 & o_y\, oz + a_y\, a_z \\ o_x\, o_z + a_x\, a_z & o_y\, o_z + a_y\, a_z & o_z^2 + a_z^2 \end{bmatrix}$$

In this case, the trace of T_k is equal to:

$$\begin{aligned} \mathrm{trace}(T_k) &= \frac{n\,\left(o_x^2 + o_y^2 + o_z^2 + a_x^2 + a_y^2 + a_z^2\right)}{\sigma^2} \\ &= \frac{n\,\left(\|\mathbf{o}\|^2 + \|\mathbf{a}\|^2\right)}{\sigma^2} \\ &= \frac{2\,n}{\sigma^2} \end{aligned}$$

This expression reflects that each of the n points of the observed edge contributes information in two d.o.f. of position. The contribution of the edge is the total contribution of all its observed points.

The orientation submatrix R_k can be written as:

$$\begin{aligned} R_k = \frac{n^2\, s^2}{\sigma^2} & \begin{bmatrix} o_x^2 + a_x^2 & o_x\, o_y + a_x\, a_y & o_x\, o_z + a_x\, a_z \\ o_x\, o_y + a_x\, a_y & o_y^2 + a_y^2 & o_y\, oz + a_y\, a_z \\ o_x\, o_z + a_x\, a_z & o_y\, o_z + a_y\, a_z & o_z^2 + a_z^2 \end{bmatrix} \\ + & \begin{bmatrix} po_x^2 + pa_x^2 & po_x\, po_y pa_x\, pa_y & po_x\, po_z + pa_x\, pa_z \\ po_x\, po_y + pa_x\, pa_y & po_y^2 + pa_y^2 & po_y\, po_z + pa_y\, pa_z \\ po_x\, po_z + pa_x\, pa_z & po_y\, po_z + pa_y\, pa_z & po_z^2 + pa_z^2 \end{bmatrix} \end{aligned}$$

The trace of this matrix is (see appendix A):

$$\begin{aligned} \mathrm{trace}(R_k) = & \frac{n^3\, s^2\, \left(o_x^2 + o_y^2 + o_z^2 + a_x^2 + a_y^2 + a_z^2\right)}{12\,\sigma^2} \\ & + \frac{n\,\left(po_x^2 + po_y^2 + po_z^2 + pa_x^2 + pa_y^2 + pa_z^2\right)}{\sigma^2} \\ = & \ \frac{n}{\sigma^2}\left(\frac{2\,n^2\, s^2}{12} + \|\mathbf{p} \times \mathbf{o}\|^2 + \|\mathbf{p} \times \mathbf{a}\|^2\right) \end{aligned}$$

$$
\begin{aligned}
&= \frac{n}{\sigma^2}\left(\frac{n^2\,s^2}{6} + 2\,\|\mathbf{p}\|^2 - \|\mathbf{p}\times\mathbf{n}\|^2\right)\\
&= \frac{n}{\sigma^2}\left(\frac{n^2\,s^2}{6} + 2\,\|\mathbf{p}\|^2 - \|\mathbf{p}\|^2\,\sin^2\alpha\right)\\
&= \frac{n}{\sigma^2}\left(\frac{n^2\,s^2}{6} + \|\mathbf{p}\|^2\left(1+\cos^2\alpha\right)\right)
\end{aligned}
$$

Thus, the contribution of the pairing in orientation depends on the relative orientation of the edge with respect to the object. When the edge has a radial orientation, we have that $\alpha = 0 \vee \alpha = \pi$, and the contribution is greater than if the edge has a transversal orientation, where $\alpha = \pm\pi/2$. ◇

In general, the trace of the position submatrix of the information matrix of a pairing accounts for the number of points contained in an observation, weighed by the number of d.o.f. in which they contribute information, and by the precision of the involved sensor. For this reason, we use the trace of T_k as a measure of the relevance of a pairing.

Definition 4.2: Relevance of a Pairing
Given a pairing p_k, whose information matrix is F_k, we define the relevance of the pairing as:

$$\texttt{relevance}(p_k) \;\triangleq\; \mathrm{trace}(T_k)$$

In the following we will describe and illustrate the most important properties of such a relevance measurement:

1. *Generality*
   This relevance measurement can be computed for any type of geometric feature and subfeature. Its covariance and binding matrices are the only elements needed to compute it. This property is very important because allows the system to evaluate the contribution to recognition of any type of feature obtained by any sensor, and to compare different types of features, such as edges and planar surfaces, contributing to make the system truly multisensor.

2. *Non-negativity*
   The T_k matrix is a positive semi-definite matrix, and thus all its eigenvalues are non-negative real numbers. Since the trace of a matrix is equal to the sum of its eigenvalues, we have that the relevance is non-negative.

3. *Dependent on Feature Dimensions*

   As we have already seen, the covariance of the observation is related to the dimensions of the observed feature, so this information will be reflected in the information matrix of the pairing. Intuitively, larger observations should be more relevant because they will have less potential pairings in the models, their location is usually more precise, and thus constraint analysis will be more discriminant.

   Example 4.6: *Sensing edges with 2D vision*
   If we use a 2D Vision system for the observation of edges, where the camera is positioned at a fixed distance from the edge, the covariance of the observation is (see chapter 2):

   $$C_E = \frac{\sigma_p^2}{n}\left[\begin{array}{ccc} 1 & -\frac{1}{d} & 0 \\ -\frac{1}{d} & \frac{1}{d^2} & 0 \\ 0 & 0 & \frac{12}{n^2 s^2} \end{array}\right]$$

   where d is the distance from the camera to the edge, s is the parameter that defines the camera resolution (mm/pixel) for a distance d, n is the number of pixels in the image that correspond to the edge, and σ_p^2 their uncertainty. The relevance of the observed 2D edge will be:

   $$\texttt{relevance}(p) = \frac{n}{\sigma_p^2}$$

   Larger edges will generate more pixels (n will be larger) and thus their relevance will be higher. ◇

4. *Dependent of Feature Symmetries*

   This is a consequence of the fact that the components of the perturbation vector which do not contribute true location information are not included in the covariance matrix of the observation. In the case of subfeatures, the binding matrix of the pairing $B_{P_k M_k}$ eliminates these components from the information matrix of the pairing. In this way, the relevance measurement takes into account how much location information the observation contributes.

   Example 4.7: *Sensing points with proximity sensors*
   We can observe different types of features with proximity sensors, for example points belonging to an edge or to a plane. Suppose that the covariance of an observation made with proximity can be modeled as:

   $$C_E = \mathrm{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2)$$

If the observation corresponds to a point on an edge we have:

$$\texttt{relevance}(p) = \frac{1}{\sigma_y^2} + \frac{1}{\sigma_z^2}$$

If it corresponds to a point on a plane:

$$\texttt{relevance}(p) = \frac{1}{\sigma_z^2}$$

This reflects the fact that points on planes give information only in the z direction, while points on edges give information in the y and z directions. ◇

5. *Dependent of Sensor Characteristics*

   This is also a consequence of the fact that sensor characteristics are reflected in the covariance matrix of the observation. Given two observations of features of the same characteristics, obtained using different sensors, the relevance of each pairing will depend on the precision of the sensor that obtained it. Again more precise observations have a smaller probability of being spuriously paired and allow more discriminant constraint analysis.

   Example 4.8: *Sensing points of a plane with proximity sensors*
   Using two proximity sensors to observe a point on a plane, where the sensors are positioned in a direction normal to the surface, and given that the uncertainty in the direction of measurement of the sensors are $\sigma_{z_1}^2$ and $\sigma_{z_2}^2$, respectively, we have:

$$\begin{aligned}\texttt{relevance}(p_1) &= \frac{1}{\sigma_{z_1}^2}\\ \texttt{relevance}(p_2) &= \frac{1}{\sigma_{z_2}^2}\end{aligned}$$

   Thus, if $\sigma_{z_1}^2 > \sigma_{z_2}^2$ (sensor 1 is less precise than sensor 2), then we have that $\texttt{relevance}(p_1) < \texttt{relevance}(p_2)$. ◇

6. *Independent of Feature Location*

   All features of the same type with the same characteristics should be equally relevant to identification. Thus, relevance should be independent of feature location. This fact can be shown as follows: from (4.4) we can express the value of F_k as:

$$F_k = J_{M_kO}^T \, A \, J_{M_kO}$$

where A is the matrix resulting from computing the corresponding expression in (4.4). The values of this matrix do not depend on the feature location, given by $\mathbf{x}_{M_kO}$. Suppose the general form of matrix A is:

$$A = \begin{bmatrix} A_p & A_c \\ A_c^T & A_o \end{bmatrix}$$

The general form of the Jacobian matrix J_{M_kO} is:

$$J_{M_kO} = \begin{bmatrix} R & S \\ 0 & R \end{bmatrix}$$

where R is the rotation submatrix of the transformation $\mathbf{x}_{M_kO}$ and $S = (\mathbf{p}\times\mathbf{n}\ \mathbf{p}\times\mathbf{o}\ \mathbf{p}\times\mathbf{a})$ (see appendix A). We have that the translation submatrix T_k of the information matrix F_k is given by:

$$T_k = R^T \, A_p \, R$$

Given that R is a unitary matrix, it does not affect the eigenvalues of A_p, and consequently, the trace of the T_k matrix remains unchanged.

7. *Additivity*
The sum of the relevances of the set of subfeatures used to estimate the location of a feature is equivalent to the relevance of the estimated feature. This is a consequence of the fact that *trace* is a linear operator, as so the trace of the information matrix of an estimation is equal to the sum of the traces of the information matrices of each observations.

Example 4.9: *Three points .vs. a plane*
Again using a proximity sensor, if we observe a point on a plane and integrate it directly to the estimation of the object location, its relevance would be:

$$\texttt{relevance}(p_k) = \frac{1}{\sigma_z^2}$$

If we observe three non-collinear such points, and estimate the location of the plane, then the relevance of the resulting plane is:

$$\texttt{relevance}(p_k) = \frac{3}{\sigma_z^2}$$

◇

4.3.2 The Relevance of an Object Location Hypothesis

The total amount of information contained in an estimation of the location of an object location hypothesis containing n pairings is given by the information matrix of the estimation Q_n:

$$Q_n = \begin{bmatrix} T_n & U_n \\ U_n^T & R_n \end{bmatrix} = \sum_{k=1}^{n} F_k = \begin{bmatrix} \sum_{k=1}^{n} T_k & \sum_{k=1}^{n} U_k \\ \sum_{k=1}^{n} U_k^T & \sum_{k=1}^{n} R_k \end{bmatrix}$$

Additivity, the last property of the relevance of a pairing, shows that the translation submatrix T_n of the information matrix of the estimation of the object location Q_n accumulates the relevance of all the pairings that have been included in the hypothesis. We will use this fact to define the relevance of an object location hypothesis.

Definition 4.3: Relevance of an Object Location Hypothesis
Given an object-location hypothesis h, composed by n observation-model pairings, whose information matrix is Q_n, we define the relevance of the hypothesis as:

$$\texttt{relevance}(h) \triangleq \text{trace}(T_n) = \sum_{k=1}^{n} \text{trace}(T_k) = \sum_{k=1}^{n} \texttt{relevance}(p_k)$$

The fundamental properties of the relevance of a hypothesis defined in this way are:

1. *Non-negative and non-decreasing*
   This is due to the fact that all eigenvalues of each T_k matrix are non-negative and add up to the relevance of the hypothesis.

2. *It is based solely on the pairings that support of the hypothesis*
   Alternative hypotheses that include the same observations in their pairings will have the same relevance.

3. *Independent of precision*
   Two hypotheses may have the same relevance and different precision. This is because precision is also related to the location of the features, while relevance is not.

4.3.3 The Relevance of an Observation

During the initial steps of the recognition process, before pairings between observations and model features are established, we must be able to select from among a set of observations the ones that may be more relevant for the identification of an object. For this purpose, we can extend the definition of relevance of a pairing taking into account that we do not know *a priori* what the values of $J_{M_k O}$ and $\hat{\mathbf{x}}_{E_k M_k}$ should be. Since the relevance of a pairing is independent of the feature location with respect to the object location, we can eliminate $J_{M_k O}$ from the equation. The relevance does depend on the value of $\hat{\mathbf{x}}_{E_k M_k}$ when we relinearize due to estimation errors, and we have that the values of $\hat{\mathbf{x}}_{E_k M_k}$ are far from zero. We will suppose that this error will be small, and thus we will take $\hat{\mathbf{x}}_{E_k M_k} = 0$. Assuming these values, from (4.4) we can calculate an *a priori* information matrix A_k:

$$A_k = B_{E_k}^T \, C_{E_k}^{-1} \, B_{E_k} \tag{4.6}$$

In the case where the observation is a subfeature, we take $\hat{\mathbf{x}}_{P_k M_k} = 0$, and from (4.5) we have:

$$A_k = B_{P_k M_k}^T \left(B_{P_k M_k} \, B_{P_k}^T \, C_{P_k} \, B_{P_k} \, B_{P_k M_k}^T \right)^{-1} B_{P_k M_k} \tag{4.7}$$

Definition 4.4: Relevance of an Observation
Given an observation e_k, with binding matrix B_{E_k}, and covariance matrix C_{E_k}, we define its relevance as:

$$\texttt{relevance}(e) \overset{\Delta}{=} \operatorname{trace}(T_k) \tag{4.8}$$

where T_k is the position submatrix of the A_k matrix calculated in (4.6). In case e_k corresponds to a subfeature, given its pairing binding matrix $B_{P_k M_k}$, its relevance is calculated from the T_k submatrix of the A_k matrix calculated in (4.7).

Example 4.10: *The relevance of a proximity edge*
Considering again an edge observed using proximity, as in example 4.3.1, from (4.6) we have:

$$F_k = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{n}{\sigma_y^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{n}{\sigma_z^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{n^3 d^2}{12\,\sigma_z^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{n^3 d^2}{12\,\sigma_y^2} \end{bmatrix}$$

Thus, the relevance of the observation is:

$$\texttt{relevance}(e) = n\left(\frac{1}{\sigma_y^2} + \frac{1}{\sigma_z^2}\right)$$

Longer edges will be more relevant (n will be greater), but also more precise edges may be more relevant (σ_y and σ_z will be smaller). ◇

4.3.4 The Potential Relevance of a Model Feature

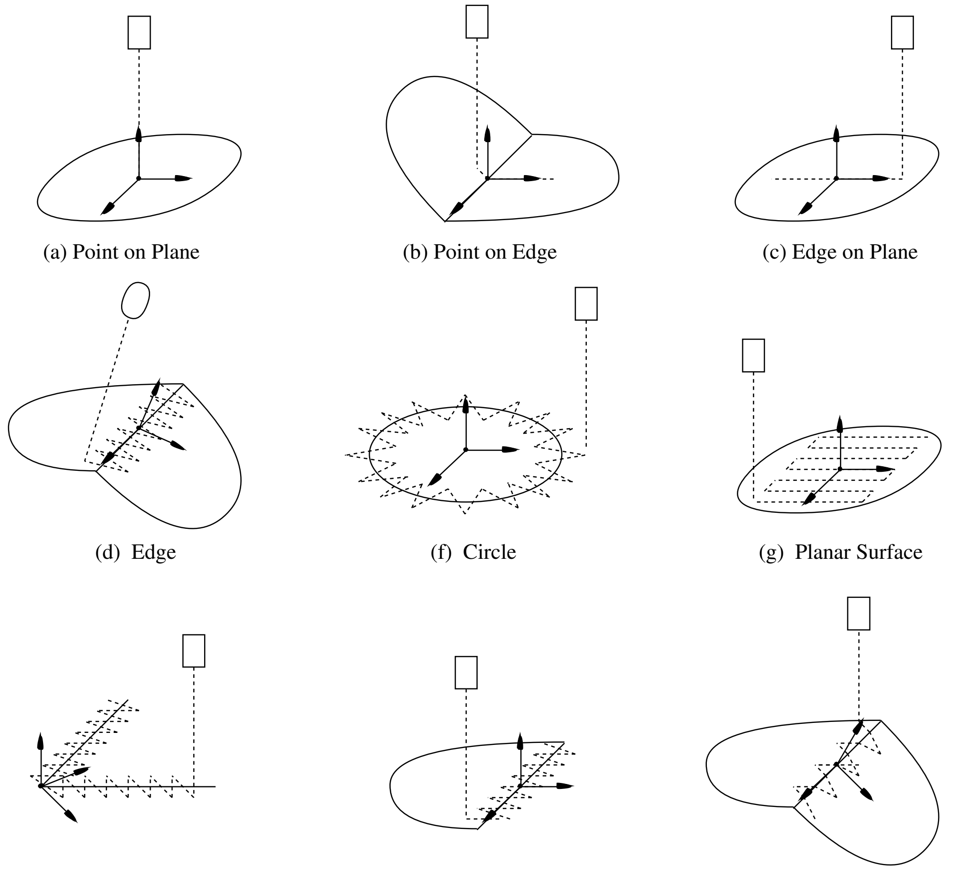
During the hypothesis verification step, the system must select a feature from the object model and verify its presence in the scene. In order to attain rapid recognition, the system needs a way of choosing among the model features the ones that will allow it to accept or reject the object location hypothesis as soon as possible. In this subsection we are interested in answering the central issue of this problem: *given an object model, how relevant is each of its model features for the recognition of the object?*

Since the contribution of a feature to recognition will depend on the relevance of the resulting observation, we can predict how relevant the model feature would be if it were perceived by some sensor. Our issue then becomes: *which type of sensor should we use in predicting the relevance of a model feature?* Without any *a priori* knowledge on which sensor may be available in the system, we use a *prototypical* sensor such as a laser proximity sensor. This type of sensor can be used to observe most types of geometric features (fig. 4.6). Furthermore, the observation strategy used to observe different geometric features is very similar, consisting basically on performing a sweeping movement along the region of interest. Thus, different types of features can be compared on a fair basis (a generic range-finder would also be suitable for this purpose). We will define the potential relevance of a model feature as the relevance that an observation of it would have, using a laser proximity sensor.

Definition 4.5: Potential Relevance of a Model Feature
Given a model feature m, we define its potential relevance by supposing we observe it completely in its predicted location with the prototypical sensor. Let e be such an observation, the potential relevance of the model feature is:

$$\texttt{potential}(m) \;\triangleq\; \texttt{relevance}(e)$$

The potential relevance for some geometric features observed using such a sensor is shown in table 4.1. Being directly related to the definition of the



(a) Point on Plane

(b) Point on Edge

(c) Edge on Plane

(d) Edge

(f) Circle

(g) Planar Surface

(g) Corner

(h) Semi-dihedral

(i) Dihedral

Figure 4.6: Observing features with laser proximity

Feature	Relevance	Parameters
Point on Plane	$\frac{1}{\sigma_z^2}$	
Point on Edge	$\frac{1}{\sigma_y^2} + \frac{1}{\sigma_z^2}$	
Line on Plane	$\frac{d}{s\,\sigma_z^2}$	$d =$ distance
Edge	$\frac{l}{s}\left(\frac{1}{\sigma_y^2} + \frac{1}{\sigma_z^2}\right)$	$l =$ edge length
Circle	$\frac{2\,\pi\,r}{s}\left(\frac{1}{\sigma_y^2} + \frac{1}{\sigma_z^2}\right)$	$r =$ circle radius
Planar surface	$\frac{a}{s^2\,\sigma_z^2}$	$a =$ surface area
Corner	$\frac{l_1+l_2}{s}\left(\frac{1}{\sigma_y^2} + \frac{1}{\sigma_z^2}\right)$	l_1, $l_2 =$ edge lengths
Semidihedral	$\frac{l}{s}\left(\frac{1}{\sigma_y^2} + \frac{2}{\sigma_z^2}\right)$	$l =$ edge length
Dihedral	$\frac{l}{s}\left(\frac{1}{\sigma_y^2} + \frac{4}{\sigma_z^2}\right)$	$l =$ edge length

Table 4.1: Potential relevance for some geometric elements (s represents the separation between observed points)

relevance of an observation, the potential relevance of a model feature is also a general measurement which fundamentally depends on feature characteristics such as dimensions. For example, the relevance of an edge fundamentally depends on its length. Additionally, such potential relevance measurement allows to compare the potential benefit of sensing different types of features.

Example 4.11: *Comparative potential relevance of an edge and a planar surface* In table 4.1 we can see that an edge of length l is more relevant than a planar surface of area a if:

$$\begin{aligned} \sigma_y^2 \gg \sigma_z^2 \quad &\Rightarrow \quad l > \frac{a}{s} \\ \sigma_y^2 \approx \sigma_z^2 \quad &\Rightarrow \quad l > \frac{a}{2\,s} \end{aligned}$$

where s is the separation between observed points. ◇

Since there may be specialized sensors which can perceive some geometric feature with higher precision than the prototypical sensor, the recognition system may maintain statistics on the average precision attained in the observation of each type of geometric feature, and modify the values of σ_y^2 and σ_z^2 accordingly. This can be considered tuning up the system using *historical* sensors for each type of feature. This tune-up will allow the system to compute the potential relevance of a feature more in accordance to its available resources.

The potential relevance can also be used to compute the *total relevance* of an object model:

Definition 4.6: Total Relevance of an Object Model
Given an object model O with k features $\{m_1, \cdots, m_k\}$ *, we define its total relevance as the sum of the potential relevance of all the features composing the model:*

$$\texttt{total}(O) \ \triangleq \ \sum_{i=1}^{k} \texttt{potential}(m_i)$$

The total relevance can give an idea of how much information can be potentially obtained for an object location hypothesis, and thus aid in deciding when an object location hypothesis may be accepted, and when it must be abandoned.

4.3.5 Relevance and the Recognition Scheme

It is important to stress that the validity of the relevance measurement greatly depends on how it is used in the context of recognition. The most important things to consider are:

- Two observations of the same model feature should not be included together to compute the relevance of the hypothesis, unless they are partial, non-overlapping observations of different regions of the feature. Otherwise, this would constitute a way of artificially increasing the relevance of the hypothesis.

- Intuitively, additional information related to regions already observed should be less relevant than information related to unobserved regions of an object. Thus, the recognition scheme should explore unobserved regions of the object, rather than trying to gather more information related to regions already observed. The objective will be attained in our system by the use of this precision measurement, which can aid the system in drawing its attention to unobserved regions of the object.

- The discrepancy between the model and observed locations is not considered. That is, it does not favor hypotheses which match better. This does not constitute a problem, because the constraint analysis allow to detect and reject potentially spurious and incoherent pairings.

The computations to obtain the proposed precision and relevance measurements are summarized in algorithm 4.1.

```
FUNCTION precision ($\mathbf{L}_{WO}$)

; $\mathbf{L}_{WO} = (\hat{\mathbf{x}}_{WO}, \hat{\mathbf{d}}_O, C_O)$: uncertain object location

; computes the precision of an object-location hypothesis

    $N :=$ diag(1, 1, 1, $r$, $r$, $r$);
    $\bar{Q} := N^{-T}\, C_O^{-1}\, N^{-1}$;
    $p :=$ min(eigenvalues($\bar{Q}$));

RETURN $p$;
END;


FUNCTION relevance ($e$)

; $e = (\mathbf{L}_{WE}, \mathcal{S}_E)$: observation

; computes the relevance of an observation
; binding matrix $B_p$ selects the position submatrix $T$

    $B_p :=$ binding $(1, 1, 1, 0, 0, 0, )$;
    $T := B_p\, B_E^T\, C_E^{-1}\, B_E\, B_p^T$;
    $r :=$ trace $(T)$;

RETURN $r$;
END;


FUNCTION potential_relevance ($m$)

; $m = (\mathbf{x}_{OM}, \mathcal{S}_M)$: observation

; computes the potential relevance of a model feature
; binding matrix $B_p$ selects the position submatrix $T$
; matrix $C_M$ is the covariance matrix of an observation of type $M$
; using the prototypical sensor

    $C_M :=$ prototypical_covariance($m$);
    $B_p :=$ binding $(1, 1, 1, 0, 0, 0, )$;
    $T := B_p\, B_M^T\, C_M^{-1}\, B_M\, B_p^T$;
    $r :=$ trace $(T)$;

RETURN $r$;
END;
```

Algorithm 4.1: *Computing Precision and Relevance*

4.4 Conclusions

In this chapter we have characterized the degree of recognition of the system with two measures associated to a hypothesis: its *precision* and its *relevance*. We have defined a precision measurement for the estimated location of an object which is related to the amount of uncertainty in the most uncertain direction in the object location. This measure allows the system to determine which sensorial observation can contribute more information in that direction. The use of such a precision measurement may imply some amount of computation, given that it is related to the problem of the determining the smallest eigenvalue of a square matrix. In some cases, like having a nearly diagonal matrix, this computation is simplified. Obtaining more precise hypotheses reduces the probability of accepting incorrect interpretations, improving the global performance of the recognition process.

We have also proposed a characterization for the relevance of a hypothesis, which is related to the characteristics of the geometric features that constitute its support, and to the sensor that observed them. This relevance measurement can be thought of as a *generalized size* measurement, which takes into account the number of d.o.f. of object location that the feature defines, and the precision of the sensor that obtained the observation. The generality of the proposed relevance measurement allows the system to compare the potential benefit of using different types of geometric features in the recognition process, and select the ones that may have a smaller number of potential pairings. Such observations also allow to perform more discriminant constraint analysis, and have a smaller probability of being spuriously paired. It must be noted that this measurement determines the relevance of a feature *per se*, that is, it does not consider the feature potential to disambiguate between similar objects. However, the computational complexity associated to performing similarity analysis on the database of object models and on the set of observations in the scene, may justify the use of the much simpler analysis proposed here.

Chapter 5

Goal-Directed Perception

Summary

This chapter is devoted to the analysis of perception tasks whose goal is the verification *of an object-location hypothesis and the* refinement *of its estimated location. We analytically characterize the* contribution *of a sensor observation as a function which relates feature characteristics, feature location, sensor capabilities and sensor location with its impact in the reduction of uncertainty in the location of the object [Neira 93b]. This characterization allows us to determine how much information can be obtained from a sensing operation, to compute the sensor location where the gain of information is maximal, and to compare the potential benefit of using different sensors for a perception task. This characterization is applicable to any type of geometric sensor and feature. We exemplify it with a comparative study of the use of mobile proximity and mobile 2D vision in the observation of straight edges.*

5.1 Introduction

One of the fundamental problems of multisensor object recognition systems consists in selecting a sensor for a given perception task, and determining its location so that the system can obtain as much information as possible from the sensing operation. This problem is denominated *purposive* or *active perception*, and has been studied from different perspectives.

A *first perspective* consists in *determining a viewpoint* from which a sensor observation would be more discriminant, in the sense of *disambiguating* between alternative object hypotheses. In [Grimson 86], Grimson works with a proximity sensor capable of measuring the three-dimensional position of a point, as well as surface normals. He proposes a procedure to determine the sensing direction most discriminant to disambiguate between a set of candidate hypotheses. Ellis [Ellis 92] extends this work by determining sensing paths, not just directions. In this work, the two-dimensional problem is studied, and some approximations to the three dimensional problem, considerably more complex, are given.

Hutchinson et al. [Hutchinson 88] propose an approach in which the system builds a viewing sphere containing the aspect graphs of all the competing hypotheses. The recognition system uses this sphere to decide which viewing point may give more discriminant information. Lee and Hahn [Lee 90] describe an optimal sensing strategy for a proximity sensor in 3D polyhedral object recognition, based on the determination of the expected number of interpretations that the perception task can prune out. In this work, uncertainty in the location of the object is not considered.

Tang and Lee [Tang 92] describe a recognition process in which the system builds a candidate discriminating graph from a set of alternative hypotheses. From this graph, the system selects the feature whose presence would allow to disambiguate between the involved hypotheses. The selection of a sensor to determine the presence of this feature is stated as a constraint satisfaction problem, in which the capabilities of each sensor are described by its set of measurable features, operational range, and working space. Thus, the selected sensor is the one whose capabilities best satisfy the set of constraints associated to the feature characteristics and location.

In [Cameron 90], a Bayesian approach to sensor planning for two degrees of freedom (d.o.f.) is proposed, in which the system determines points of the scene whose observation with an ideal sensor would give more discriminant information. Its generalization to the six d.o.f., feature-based, multisensor case seems complex.

From a *second perspective*, efforts have been concentrated not on obtaining more discriminant information, but on obtaining information that allow complete scene reconstruction. Some researchers have studied the use of *specific sensor combinations* in the recognition of objects. Allen [Allen 87] uses two sensors that are complementary in nature: vision and touch. Due to the global range of the vision sensor, it is used to acquire information for the generation of hypotheses. Information required for the verification is these hypotheses is obtained by the touch sensor that explores regions of the object that are occluded to vision.

Chiou et al. [Chiou 92] investigate the characteristics of the measurement error for different configurations of two cameras, in order to determine which camera locations are more suitable to estimate more precisely the object locations. Maver and Bajcsy [Maver 93] study the more general *next view problem* in which the system concentrates its attention in observing regions of the scene which have been occluded in previous views.

In summary, the goal of the first group of approaches is to disambiguate between several competing hypotheses, while the goal of the second group is to verify an object-location hypothesis. Most of the work done in this area is concerned with one type of geometric feature and considers one class of sensors. An additional limitation of most of these approaches is that uncertainty is modeled using maximal error bounds, which can lead to conservative estimations of uncertainty.

We study the problem from the second perspective. Nevertheless, we are interested not only in *verifying the identity* of the object, but also in *reducing the location uncertainty* of the hypothesis. We use the term *goal-directed perception* in the context of perception tasks that can make use of *a priori* information about the location of a feature required by the recognition system [Neira 93b]. The goal of the perception task is to verify the presence of the feature in the scene, and to reduce the uncertainty in the object location.

We propose a procedure to select a model feature whose observation would reduce more uncertainty in object location. We also propose a procedure to select a sensor and compute its location so that the reduction of uncertainty in observing this feature is maximal. We explicitly consider information related to the involved geometric feature (its geometrical properties and range of visibility), as well as sensor capabilities (its precision and admissible range of locations). We consider all types of geometric features and sensors, allowing the system to be truly multisensor.

This chapter is organized as follows: section 5.2 deals with the characterization of the contribution of sensor observations in reducing the uncertainty in the location of the object. This characterization is used in sections 5.3 and

5.4 to study the use of mobile proximity and mobile 2D vision in the observation of straight edges. Section 5.5 contains a description of the procedure to select the most contributing feature. In section 5.6 we use an example to illustrate how the contribution measure can be used to compare the use of these two sensors in the verification of an object location hypothesis.

5.2 Characterizing the Contribution of an Observation

Recall from the preceding chapter that during the integration process, the information matrix of the estimation Q_n accumulates location information contributed by the sensor observations, and that the location information contributed by each observation is contained in the information matrix of the pairing F_k. Consider an observation e_k, whose uncertain location is represented by $\mathbf{L}_{WE_k} = (\hat{\mathbf{x}}_{WE_k}, \hat{\mathbf{p}}_{E_k}, C_{E_k})$, and a model feature m_k, whose location in the scene according to the hypothesis is given by $\mathbf{L}_{WM_k} = (\hat{\mathbf{x}}_{WM_k}, \hat{\mathbf{p}}_{M_k}, C_{M_k})$. In chapter 2 we have seen from (2.8) that this information matrix F_k is calculated as :

$$F_k = H_k^T (G_k S_k G_k^T)^{-1} H_k$$

In the case of the integration of a feature to estimate the location of an object, from (2.11) we have:

$$\begin{aligned} H_k &= (B_{E_k} J_{2\oplus}\{\hat{\mathbf{x}}_{E_k M_k}, 0\})\, J_{M_k O} \\ G_k &= -B_{E_k} J_{1\oplus}\{0, \hat{\mathbf{x}}_{E_k M_k}, \} B_{E_k}^T \\ S_k &= C_{E_k} \end{aligned} \tag{5.1}$$

Thus, F_k is calculated as follows:

$$\begin{aligned} F_k = \; & (B_{E_k} J_{2\oplus}\{\hat{\mathbf{x}}_{E_k M_k}, 0\}\, J_{M_k O})^T \\ & (B_{E_k} J_{1\oplus}\{0, \hat{\mathbf{x}}_{E_k M_k}, \} B_{E_k}^T C_{E_k} B_{E_k} J_{1\oplus}^T\{0, \hat{\mathbf{x}}_{E_k M_k}, \} B_{E_k}^T)^{-1} \\ & (B_{E_k} J_{2\oplus}\{\hat{\mathbf{x}}_{E_k M_k}, 0\}\, J_{M_k O}) \end{aligned} \tag{5.2}$$

where $J_{1\oplus}$ and $J_{2\oplus}$ are the Jacobians of the composition of location vectors (see appendix A). Similar expressions can be obtained for the case where e_k represents an observed subfeature. As we have seen, the fundamental factors that are involved in the calculation of this matrix are the *characteristics of*

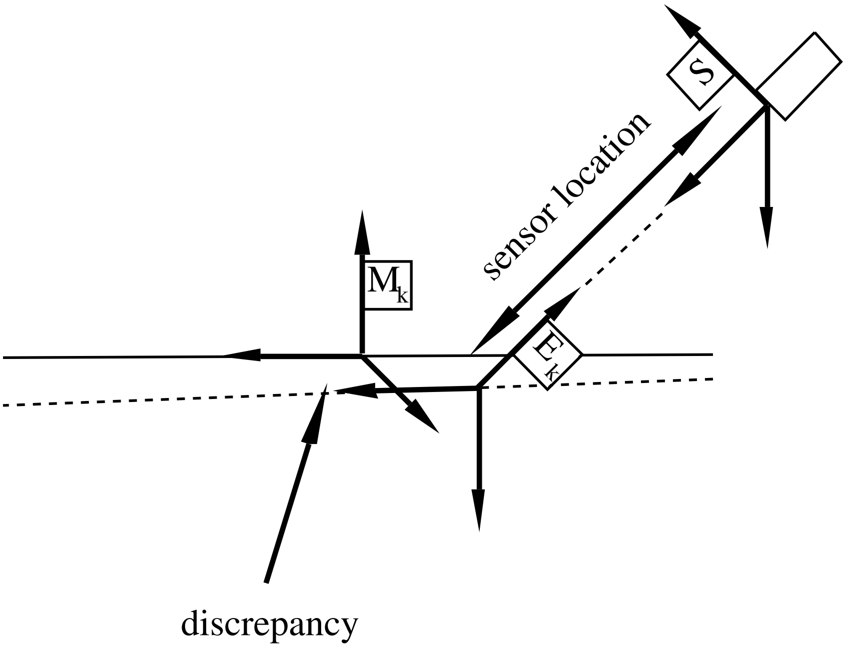


Figure 5.1: The relative location vector $\mathbf{x}_{E_k M_K}$ describes the discrepancy between the observation and the model feature, and the location of the sensor relative to the model feature

the feature (represented by B_{E_k}), its *location relative to the object location* (represented by $J_{M_k O}$), and the *characteristics of the sensor and the sensing strategy* (represented by the covariance matrix C_{E_k}). Additionally, we can see that the relative location of the observation with respect to the predicted location of the model feature, represented by $\hat{\mathbf{x}}_{E_k M_k}$, also influences the values of the F_k matrix. This location vector represents two things (fig. 5.1):

- The discrepancy between the predicted location of the feature m_k and the observation e_k.

- The location of the sensor relative to the model feature. This happens because we choose to align the reference of the observation with the corresponding sensor reference. Thus, if the observation reference is *aligned* with the sensing direction, this reference changes if the sensor varies its location.

In our recognition scheme, the location of the object is estimated as soon as possible, using the most relevant features, which usually have good precision. Thus, prediction errors are normally small. This will make the discrepancy between the predicted location of the feature and its corresponding observation small. Therefore, we will concentrate on the study of $\hat{\mathbf{x}}_{E_k M_k}$ as a function of the sensor location.

The contribution of location information of a pairing is given by the eigenvalues of its information matrix F_k. Given that the trace of a matrix is equal to the sum of its eigenvalues, we can determine how the observation will contribute to the object location by analyzing the diagonal elements of the F_k matrix.

Definition 5.1: Contribution of a Pairing to the Position and Orientation of an Object-Location Hypothesis
Given an observation model pairing p_k between an observation e_k and a model feature m_k, whose information matrix F_k is calculated as above, and has the general form:

$$F_k = \left[\begin{array}{cc} T_k & U_k \\ U_k^T & R_k \end{array} \right]$$

we define the contribution *of the pairing to each of the components of the object location vector as:*

$$\begin{array}{rcl}
\texttt{contribution}^x(p_k) & \triangleq & T_k[1,1] \\
\texttt{contribution}^y(p_k) & \triangleq & T_k[2,2] \\
\texttt{contribution}^z(p_k) & \triangleq & T_k[3,3] \\
\texttt{contribution}^\psi(p_k) & \triangleq & R_k[1,1] \\
\texttt{contribution}^\theta(p_k) & \triangleq & R_k[2,2] \\
\texttt{contribution}^\phi(p_k) & \triangleq & R_k[3,3]
\end{array}$$

The total contribution of the pairing in position and orientation is then given by:

$$\begin{array}{rcl}
\texttt{contribution}^P(p_k) & \triangleq & \mathbf{trace}(T_k) \\
\texttt{contribution}^O(p_k) & \triangleq & \mathbf{trace}(R_k)
\end{array}$$

The are two important observations to be made with respect to this contribution measurement:

- Feature contribution is closely related to feature *relevance*. The total contribution of a feature to the position parameters of the object location is equal to the relevance of the observation. As we will see in this chapter, the total contribution in position *does not* depend on the sensor location. This means that the sensor location does not affect the relevance of the observation. The feature contribution concept simply allows us to analyze the location information that the feature contributes *on each* axis. If we locate the sensor so that its contribution along one axis is greater, its contribution along the others will be smaller.

- We are not considering the elements of the information matrix F_k outside of the diagonal, which express the coupling between the components of the location vector of the object. This means that we are not taking into account the potential of the observation in reducing this coupling. Thus, our contribution is a conservative measurement. This fact has limited consequences because the *precision measurement* always directs the system to explore unobserved regions of the object, and if we obtain observations from different regions of the object, the hypothesis will be balanced and the coupling will be small.

Given an object location hypothesis, whose estimated location is given by $\mathbf{L}_{WO}$, and given a model feature, whose estimated location according to the hypothesis is given by $\mathbf{L}_{WM_k}$, we propose a procedure to select the sensor and computing the sensor location where the observed feature will contribute more location information. This procedure consists in predicting the contribution of the observation to reduce the uncertainty of object location, supposing there will be no prediction error in the location of the feature.

Consider the use of some sensor S to verify the location of this feature. Let $\mathbf{L}_{WE_k}$ express the location of the observed feature obtained by this sensor. (The observation reference is aligned with the sensing direction.) Given that the contribution is a function of the relative transformation between the location of the observation and of the prediction $\hat{\mathbf{x}}_{E_kM_k}$, we can determine for which values of $\hat{\mathbf{x}}_{E_kM_k}$ the contribution of F_k is maximal.

The admissible values for $\hat{\mathbf{x}}_{E_kM_k}$ are restricted by:

- The pairing relationship between the observation and the model feature, corresponding to the direct or inverse constraint. In case e_k corresponds to a feature, we have:

  $$B_{E_k}\,\mathbf{x}_{E_kM_k} = 0$$

  If there is no prediction error, then we also have:

  $$B_{E_k}\,\hat{\mathbf{x}}_{E_kM_k} = 0$$

  Thus, $\hat{\mathbf{x}}_{E_kM_k}$ must belong to the set of symmetries of the feature. In case e_k corresponds to a subfeature, we have:

  $$B_{E_kM_k}\,\mathbf{x}_{E_kM_k} = 0$$

  Again, if there is no prediction error, then we also have:

  $$B_{E_kM_k}\,\hat{\mathbf{x}}_{E_kM_k} = 0$$

  Thus, $\hat{\mathbf{x}}_{E_kM_k}$ must belong to the set of symmetries of the pairing relationship.

- Constraints on the sensor location due to sensor characteristics. For example, proximity sensors have a limited range of observation and they must be positioned in directions near to the normal of the observed surface.

- The range of visibility of the feature. The geometry of the object restricts the directions from which the feature is observable.

In the next sections we exemplify the proposed sensor and sensing modality selection method by showing how to select a sensor and computing its location to verify the predicted location of an edge. Although the analysis of perception tasks described here is applicable to any type of geometric sensor, for illustrative purposes we will consider the use of the two types of sensors modeled in chapter 2: a proximity sensor mounted on the robot finger, and a 2D vision system that uses a camera mounted on the robot hand. The comparative study of these two sensors is interesting for the following reasons:

- Proximity edges give information on two d.o.f. of position, while 2D vision gives information in only one.
- 2D vision observes more points (pixels) than it is normally feasible with proximity.
- Proximity can be considered more reliable than 2D vision. Since it measures distance, its more close to the object topology. 2D vision can give spurious observations caused by shadows and other undesired illumination effects.
- The precision in position that can be obtained with proximity is higher because the errors due to the gripper positioning for the obtention of each measured point tend to cancel each other, given that the robot is appropriately calibrated and there are no systematic positioning errors. This is not the case in 2D vision, where all pixels in an image are affected by the same positioning error of the gripper.
- The proximity sensor is constrained to be positioned in a direction nearly normal to the surface of the face, in order to assure that the echo signal will be received. On the other hand, the vision sensor can obtain observations from any direction within the range of visibility of the feature.
- 2D vision can be considered faster than proximity because it involves only the camera positioning movement, while proximity performs a sweeping movement of the sensor along the edge. In this work are not concerned with sensing costs. Thus, we will study sensing operations that can be considered equivalent in cost.

In the next two sections we will analyze the measurement of contribution as a function of sensor characteristics and sensor location for these two types of sensors. Then we will present a comparative example in which the system uses this information to select the sensor and computing the sensor location that are best suited for a specified perception task.

5.3 Mobile Proximity

5.3.1 Computing the Contribution of a Proximity Edge

Suppose we use the proximity sensor to obtain an observation of an edge. Recall from chapter 2, equation (2.16), that the covariance of an observation of this type is:

$$C_{E_k} = \begin{bmatrix} \frac{\sigma_y^2}{n} & 0 & 0 & 0 \\ 0 & \frac{\sigma_z^2}{n} & 0 & 0 \\ 0 & 0 & \frac{12\,\sigma_z^2}{n^3 s^2} & 0 \\ 0 & 0 & 0 & \frac{12\,\sigma_y^2}{n^3 s^2} \end{bmatrix}$$

where n is the number of observed points, and s is the distance between successive points. The values of σ_y and σ_z describe both the sensor and robot precisions in those directions. In determining the expected value of the relative location between the observation and the predicted model feature we make two considerations:

- Given that we have *a priori* information, that is, the predicted location of the model feature, we shall locate the sensor so that its sensing direction is normal to the edge (figure 5.2).
- We will locate the sensor at a fixed distance from the edge. The values of σ_y and σ_z decrease if we reduce this distance, but the sensor has a limited range of measurement. Also, the risk of collisions must be avoided.
- Supposing there is no prediction error, the relative transformation between the observation and the prediction must belong to the set of symmetries of the feature. Considering that we will completely observe the edge, we place the reference of the observed feature E_k in the center of the edge, aligned with the sensing direction:

  $$\hat{\mathbf{x}}_{E_k M_k} = (0,\ 0,\ 0,\ \lambda,\ 0,\ 0)^T$$

  where λ expresses the relative orientation of the sensor with respect to the model feature reference M_k.

The self-binding matrix of edges is:

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

Thus, to calculate the information matrix F_k, from (5.1) we have:

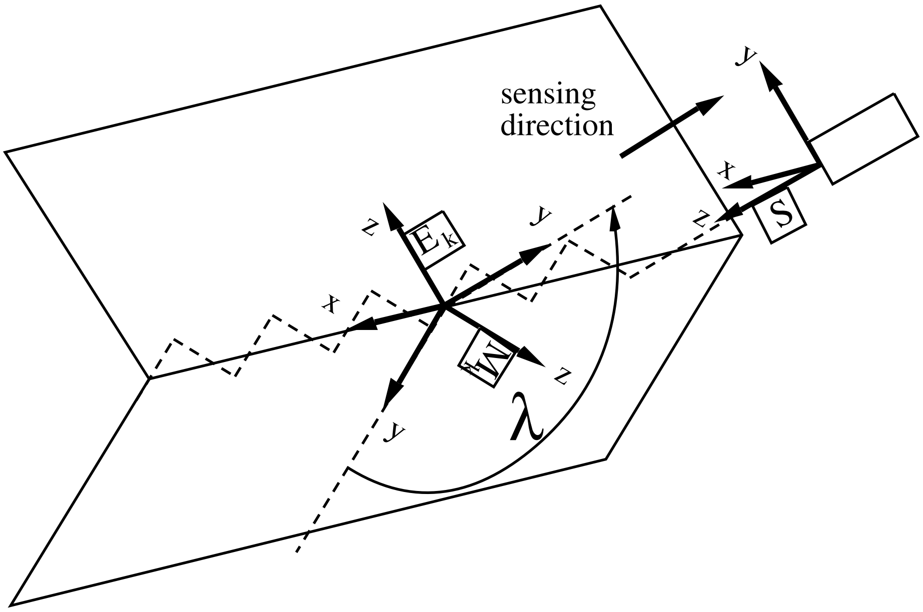


Figure 5.2: Observing an edge using proximity

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

$$H_k = \begin{bmatrix} o_x\,C\lambda - a_x\,S\lambda & o_y\,C\lambda - a_y\,S\lambda & o_z\,C\lambda - a_z\,S\lambda & po_x\,C\lambda - pa_x\,S\lambda & po_y\,C\lambda - pa_y\,S\lambda & po_z\,C\lambda - pa_z\,S\lambda \\ o_x\,S\lambda + a_x\,C\lambda & o_y\,S\lambda + a_y\,C\lambda & o_z\,S\lambda + a_z\,C\lambda & po_x\,S\lambda + pa_x\,C\lambda & po_y\,S\lambda + pa_y\,C\lambda & po_z\,S\lambda + pa_z\,C\lambda \\ 0 & 0 & 0 & o_x\,C\lambda - a_x\,S\lambda & o_y\,C\lambda - a_y\,S\lambda & o_z\,C\lambda - a_z\,S\lambda \\ 0 & 0 & 0 & o_x\,S\lambda + a_x\,C\lambda & o_y\,S\lambda + a_y\,C\lambda & o_z\,S\lambda + a_z\,C\lambda \end{bmatrix}$$

where $S\lambda = \sin(\lambda)$ and $C\lambda = \cos(\lambda)$, and o_x, o_y, o_z, a_x, a_y, a_z, po_x, po_y, po_z, pa_x, pa_y, and pa_z are the components of the Jacobian J_{M_kO} (see appendix

A). In evaluating which value of λ maximizes this contribution, we make the following considerations:

- Let $[\lambda_m, \lambda_M]$ denote the range of visibility of the edge (considering only the local geometry of the object). We must select a sensing direction from which the edge is visible, that is $\lambda \in [\lambda_m, \lambda_M]$.
- We are constrained to locate the proximity sensor in a direction near to the normal of the sensed surface. In the case of an edge, this means that there are only two possible ranges of values for λ, corresponding to the normals of the two adjacent faces. Let $\lambda_1 \pm \epsilon$ and $\lambda_2 \pm \epsilon$ denote them (in our case the value of ϵ can be around 10^o).

The selected direction λ will be the admissible sensor location within the range of visibility from which the contribution is maximal. In the next paragraphs we will analyze the algebraic expressions for the contribution of an observed edge in position and in orientation, without taking into account such constraints. These will be incorporated to the analysis when comparing the contribution of this sensor with mobile 2D vision.

Table 5.1 summarizes the expressions of the contribution functions on each axis. In the following we analyze the expressions of these contribution functions in position and in orientation.

Contribution to Position in one Axis

Consider the expression for the contribution of the edge along the x axis, given in table 5.1. We can see that two factors determine the value of λ for which the contribution of the observation to the object location along the x axis is greater:

1. *Sensor Characteristics:* the values of σ_y and σ_z determine which location contributes more information (fig. 5.3). If the sensor gives equal precision in axes y and z ($\sigma_y = \sigma_z = \sigma$), we have:

$$\texttt{contribution}^x\,(p_k) = n\,\frac{a_x^2 + o_x^2}{\sigma^2}$$

   In this case the contribution does not depend on λ, so any sensor location will contribute the same information to the x component of the object location vector.

axis	`contribution`
x	$\frac{n(o_x\,C\lambda - a_x\,S\lambda)^2}{\sigma_y^2} + \frac{n(o_x\,S\lambda + a_x\,C\lambda)^2}{\sigma_z^2}$
y	$\frac{n(o_y\,C\lambda - a_y\,S\lambda)^2}{\sigma_y^2} + \frac{n(o_y\,S\lambda + a_y\,C\lambda)^2}{\sigma_z^2}$
z	$\frac{n(o_z\,C\lambda - a_z\,S\lambda)^2}{\sigma_y^2} + \frac{n(o_z\,S\lambda + a_z\,C\lambda)^2}{\sigma_z^2}$
P	$n\left(\frac{1}{\sigma_y^2} + \frac{1}{\sigma_z^2}\right)$
ψ	$\frac{n}{\sigma_y^2}\left((po_x\,C\lambda - pa_x\,S\lambda)^2 + n^2\,s^2\frac{(o_x\,S\lambda + a_x\,C\lambda)^2}{12}\right)$ $+\frac{n}{\sigma_z^2}\left((po_x\,S\lambda + pa_x\,C\lambda)^2 + n^2\,s^2\,\frac{(o_x\,C\lambda - a_x\,S\lambda)^2}{12}\right)$
θ	$\frac{n}{\sigma_y^2}\left((po_y\,C\lambda - pa_y\,S\lambda)^2 + n^2\,s^2\frac{(o_y\,S\lambda + a_y\,C\lambda)^2}{12}\right)$ $+\frac{n}{\sigma_z^2}\left((po_y\,S\lambda + pa_y\,C\lambda)^2 + n^2\,s^2\,\frac{(o_y\,C\lambda - a_y\,S\lambda)^2}{12}\right)$
ϕ	$\frac{n}{\sigma_y^2}\left((po_z\,C\lambda - pa_z\,S\lambda)^2 + n^2\,s^2\frac{(o_z\,S\lambda + a_z\,C\lambda)^2}{12}\right)$ $+\frac{n}{\sigma_z^2}\left((po_z\,S\lambda + pa_z\,C\lambda)^2 + n^2\,s^2\,\frac{(o_z\,C\lambda - a_z\,S\lambda)^2}{12}\right)$
O	$\frac{n\left((n^2 s^2 + 12\lVert\mathbf{p}\rVert^2)\,(\sigma_y^2 + \sigma_z^2) - 12\,\lVert\mathbf{p}\rVert^2\,f(\lambda)\right)}{12\,s\,\sigma_y^2\,\sigma_z^2}$ $f(\lambda) = (\cos\beta\,\sin\lambda + \cos\gamma\,\cos\lambda)^2\,\sigma_y^2$ $+(\cos\beta\,\cos\lambda - \cos\gamma\,\sin\lambda)^2\,\sigma_z^2$

Table 5.1: Contribution functions for an edge observed using the proximity sensor; n: number of observed points; s: separation between points; σ_y and σ_z: sensor precision; λ: sensing direction; o_x, o_y, o_z, a_x, a_y, a_z, po_x, po_y, po_z, pa_x, pa_y, and pa_z: components of the Jacobian matrix J_{M_kO}; $\mathbf{p}$: position vector of $\mathbf{x}_{OM_k}$; β and γ: angles between the position vector $\mathbf{p}$ and unit vectors $\mathbf{o}$ and $\mathbf{a}$ (see appendix A)

2. *Feature Location:* the contribution of the observation is related to the values of o_x and a_x, which determine the edge orientation relative to the x axis of the object reference. Since $n_x^2 + o_x^2 + a_x^2 = 1$ (see appendix A), the contribution is maximal when $n_x^2 = \cos^2\phi\cos^2\theta = 0$. This happens when $\theta = \pm\pi/2$ or $\phi = \pm\pi/2$, that is, when the edge is orthogonal to the x axis of the object reference (fig. 5.4.a). On the other hand, the contribution is minimal when $n_x^2 = 1$, that is, $\theta = \phi = 0$. In this case, the edge is parallel to the x axis of the object reference (fig. 5.4.b) In this case `contribution`$^x\,(p_k) = 0$ for any value of λ. This means that the edge location *cannot* contribute information

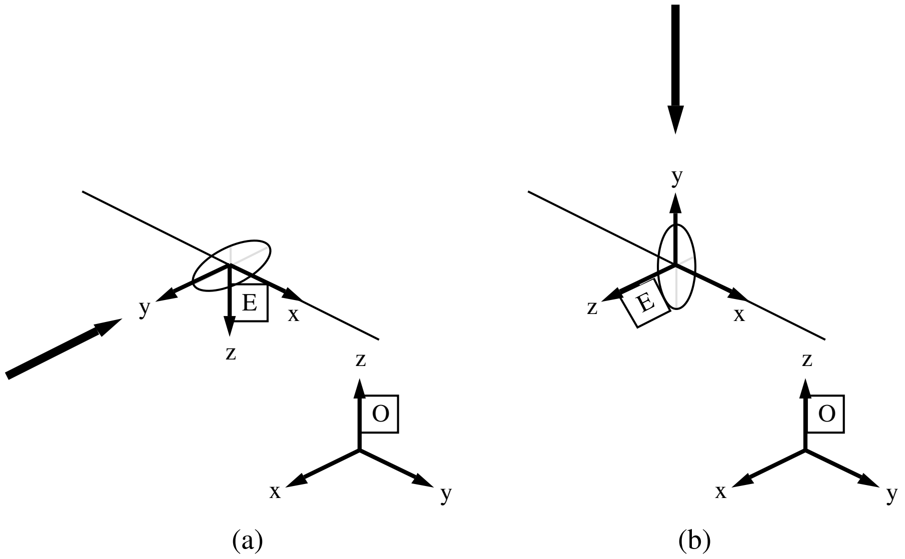


Figure 5.3: Contribution of an observed edge along the x axis of the object reference O when $\sigma_y > \sigma_z$. (a) minimal when most uncertain direction of the sensor (axis y of the feature) is aligned with the x axes of the object reference; (b) maximal when most precise direction of the sensor (axis z of the feature) is aligned with the x axes of the object reference

in the direction of the x axis.

Total Contribution in Position

The contribution of the edge to the position of the object along all the axes is equal to the trace of the position submatrix T_k of the information matrix of the estimation The result is given the P row of table 5.1. We can see that the contribution to position does not depend on λ. Thus, to improve the estimated position of the object, any sensor location is equivalent. Its contribution is fundamentally related to the length of the edge—which depends on n—and to the sensor precision (represented by σ_y^2 and σ_z^2). Note also that this contribution is independent of the relative location of the feature with respect to the object. This agrees with the intuitive idea that any point of the object contributes equivalently to the estimation of position.

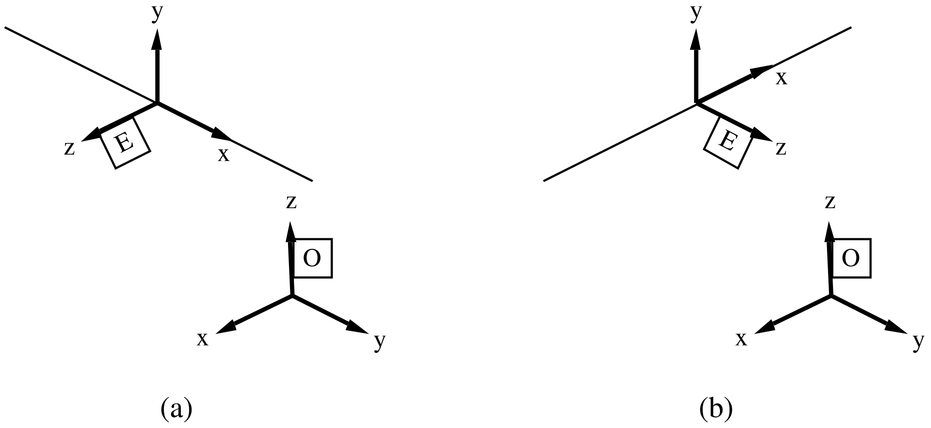


Figure 5.4: The contribution of an observed edge along the x axis (a) is maximal when the edge is orthogonal to the axis; (b) is zero when the edge is parallel to the axis

Contribution to Orientation in one Axis

Let us analyze how the observation of an edge using proximity contributes to an orientation component of the estimation of the object location. In the case of the ψ component (see table 5.1), we can see again that two fundamental factors have influence in the value the contribution around the x axis:

1. *Sensor Characteristics:* the values of σ_y and σ_z condition which orientation renders the maximal and minimal contributions (fig. 5.5). when $\sigma_y = \sigma_z = \sigma$, the contribution is given by:

$$\texttt{contribution}^{\psi}(p_k) = \frac{n^3 s^2 \left(\frac{o_x^2}{12} + \frac{a_x^2}{12}\right) + n\left(po_x^2 + pa_x^2\right)}{\sigma^2}$$

   In this case the contribution is independent of the value of λ, so any sensor orientation contributes the same information.

2. *Feature Location:* The contribution of the edge depends on its relative position (represented by po_x and pa_x) and its relative orientation (represented by o_x and a_x). When the edge is aligned with the x axis

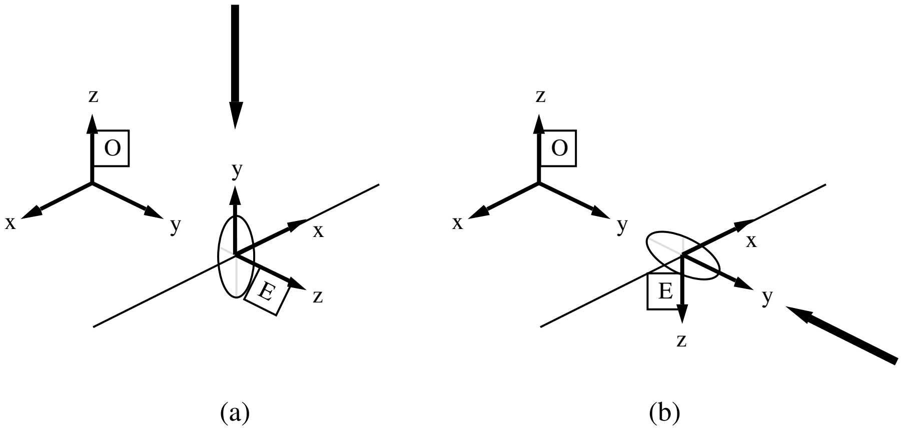


Figure 5.5: The contribution of an observed edge to the object location around the x axis when $\sigma_y > \sigma_z$: (a) is minimal; (b) is maximal

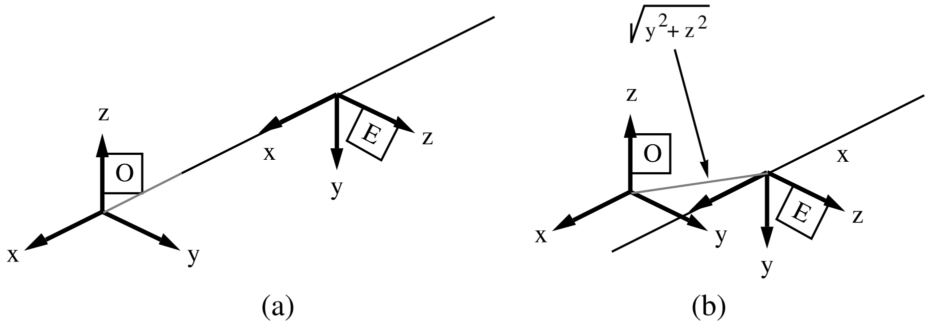


Figure 5.6: The contribution around the x axis depends on feature location; (a) it is equal to zero when the edge is aligned with the axis; (b) it depends on relative distance when the edge is parallel to the axis

($y = z = \theta = \phi = 0$), its contribution to orientation becomes zero (fig. 5.6.a). When the edge is parallel to the x axis, its contribution depends on its distance to the edge (fig. 5.6.b):

$$\texttt{contribution}^{\psi}(p_k) = \frac{n\,(y^2 + z^2)}{\sigma^2}$$

Total Contribution to Orientation

The total contribution to orientation, calculated as the trace of the orientation submatrix R_k of the information matrix of the observation, is given in row O of table 5.1. Let us analyze the two factors involved in this contribution:

1. *Sensor Characteristics:* again which solution is maximal depends on the values of σ_y and σ_z. When $\sigma_y = \sigma_z = \sigma$, the contribution becomes:

$$\texttt{contribution}^{O}(p_k) = \frac{n}{6\,\sigma^2}\left(n^2\,s^2 + 6\|\mathbf{p}\|^2(1 + \cos^2\alpha)\right)$$

   where α is the angle between $\mathbf{p}$ and $\mathbf{n}$ (see appendix A). In this case the value of the contribution is independent of λ and only depends on the location of the edge.

2. *Feature Location:* edges more distant to object location contribute more to orientation, but the orientation of the edge also influences this value, specifically the value of $f(\lambda)$. When the edge has transversal orientation ($\mathbf{p} \perp \mathbf{n}$), we have $\cos\alpha = 0$, and thus $f(\lambda) = \sigma^2$ (fig. 5.7.a). When the edge has a radial orientation ($\mathbf{p}\|\mathbf{n}$), we have $\cos\beta = \cos\gamma = 0$, and thus $f(\lambda) = 0$ (fig. 5.7.b)

We can draw three general conclusions from this analysis:

1. The sensor orientations where the contributions are minimal and maximal are *orthogonal*. This is true both in the case of mobile proximity and 2D vision.

2. Features whose location is optimal for contributing location information along one axis, are those whose location is less suitable for contributing location information along another.

3. The sensor orientation where the contribution along one axis is maximal, is the sensing direction where the contribution along another is minimal.

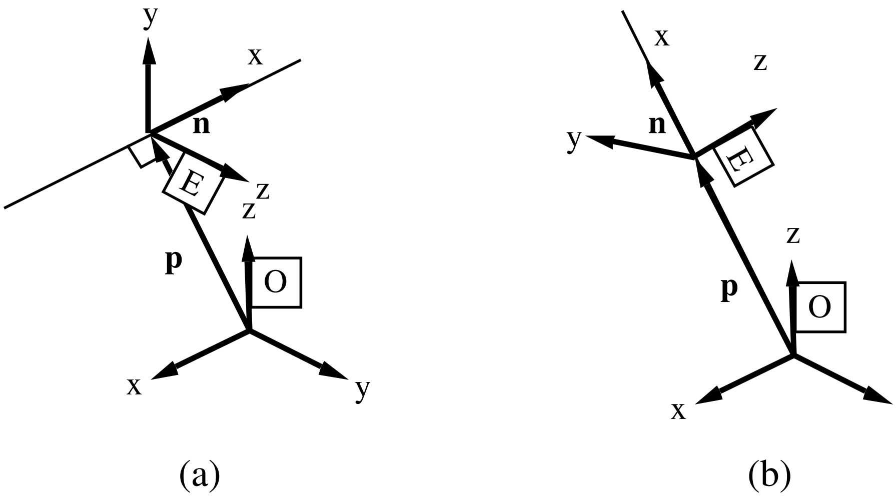


Figure 5.7: The total contribution to orientation depends on feature location; (a) it is *minimal* when the edge is normal to vector **p**; (b) it is *maximal* when the edge is aligned with vector **p**

5.3.2 Computing the Contribution of a Point on an Edge

The complete observation of an edge can be quite expensive (especially with proximity sensors), and in some cases the observation of only one point of the edge gives enough information to refine the estimation of the object location. Let us first study the contribution that a point on an edge observed with proximity can make to the location of the object. Let P_k be a reference associated to the observed point, whose estimated location is represented by $\mathbf{L}_{WP_k} = (\hat{\mathbf{x}}_{WP_k}, \hat{\mathbf{p}}_{P_k}, C_{P_k})$ (fig. 5.8). From (2.15) we have that the covariance associated to an observation of this type is (chapter 2):

$$C_{P_k} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & 0 \\ \sigma_{xy} & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}$$

Supposing there is no prediction error, the relative location between the edge and the observed point would be:

$$\hat{\mathbf{x}}_{M_kP_k} = (r, 0, 0, \lambda, 0, 0)^T$$

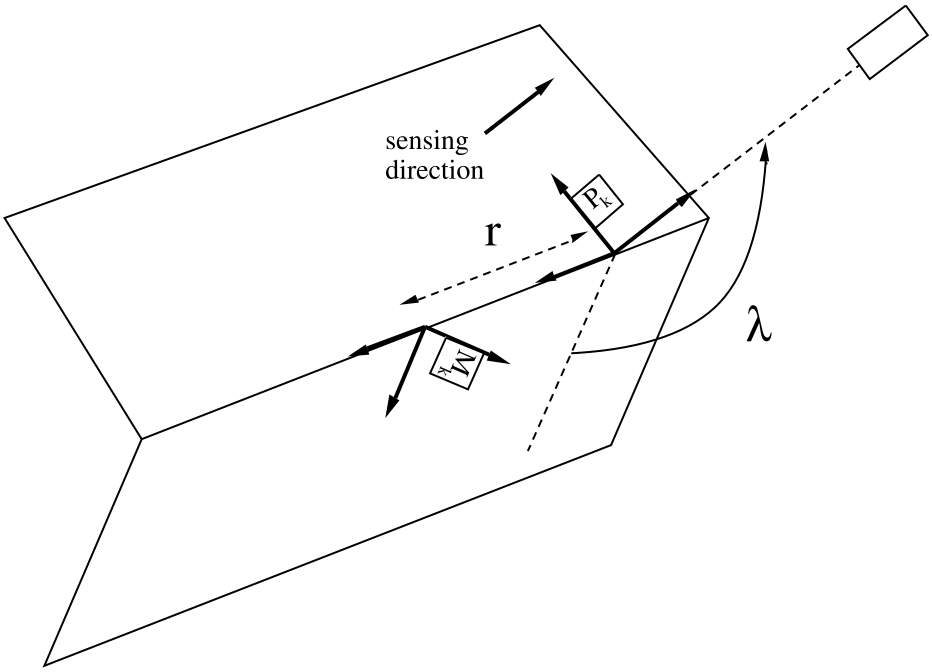


Figure 5.8: Observing a point on an edge using proximity

where r is the distance from the point to the reference of the edge, and λ represents the relative orientation of the sensor with respect to the edge. The pairing relationship between the edge and a point belonging to it is described by the following binding matrix:

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

Thus, from (5.1) we have:

$$G_k = \begin{bmatrix} 0 & C\lambda & -S\lambda \\ 0 & S\lambda & C\lambda \end{bmatrix}$$

$$H_k = \begin{bmatrix} o_x & o_y & o_z & po_x + r\,a_x & po_y + r\,a_y & po_z + r\,a_z \\ a_x & a_y & a_z & pa_x - r\,o_x & pa_y - r\,o_y & pa_z - r\,o_z \end{bmatrix}$$

The contribution of the observation as a function of the sensor orientation λ gives equal results—in terms of determining the optimal sensor

axis	`contribution`
x	$\frac{(o_x\,S\lambda - a_x\,C\lambda)^2}{\sigma_z^2} + \frac{(o_x\,C\lambda + a_x\,S\lambda)^2}{\sigma_y^2}$
y	$\frac{(o_y\,S\lambda - a_y\,C\lambda)^2}{\sigma_z^2} + \frac{(o_y\,C\lambda + a_y\,S\lambda)^2}{\sigma_y^2}$
z	$\frac{(o_z\,S\lambda - a_z\,C\lambda)^2}{\sigma_z^2} + \frac{(o_z\,C\lambda + a_z\,S\lambda)^2}{\sigma_y^2}$
P	$\frac{1}{\sigma_y^2} + \frac{1}{\sigma_z^2}$
ψ	$\frac{((po_x + r\,a_x)S\lambda + (r\,o_x - pa_x)C\lambda)^2}{\sigma_z^2} + \frac{((r\,o_x - pa_x)S\lambda - (r\,a_x + po_x)C\lambda)^2}{\sigma_y^2}$
θ	$\frac{((po_y + r\,a_y)S\lambda + (r\,o_y - pa_y)C\lambda)^2}{\sigma_z^2} + \frac{((r\,o_y - pa_y)S\lambda - (r\,a_y + po_y)C\lambda)^2}{\sigma_y^2}$
ϕ	$\frac{((po_z + r\,a_z)S\lambda + (r\,o_z - pa_z)C\lambda)^2}{\sigma_z^2} + \frac{((r\,o_z - pa_z)S\lambda - (r\,a_z + po_z)C\lambda)^2}{\sigma_y^2}$
O	$\frac{\left(f(r) - \lVert\mathbf{p}\rVert^2\,g(\lambda)\right)}{\sigma_y^2\,\sigma_z^2}$ $f(r) = (r^2 + 2\,r\,\lVert\mathbf{p}\rVert\ \cos\alpha + \lVert\mathbf{p}\rVert^2)\,(\sigma_y^2 + \sigma_z^2)$ $g(\lambda) = (\cos\beta\sin\lambda + \cos\gamma, \cos\lambda)^2\sigma_y^2 + (\cos\beta\cos\lambda - \cos\gamma\sin\lambda)^2\sigma_z^2$

Table 5.2: Contribution functions for a point on an edge observed using the proximity sensor; r: distance of the point with respect to the edge reference; σ_y and σ_z: sensor precision; λ: sensing direction; o_x, o_y, o_z, a_x, a_y, a_z, po_x, po_y, po_z, pa_x, pa_y, and pa_z: components of the Jacobian matrix J_{M_kO}; $\mathbf{p}$: position vector of $\mathbf{x}_{OM_k}$; β and γ: angles between the position vector $\mathbf{p}$ and unit vectors $\mathbf{o}$ and $\mathbf{a}$ (see appendix A)

orientation—to those of the preceding subsection. Thus, our analysis will concentrate in the sensor position, determined by r. The expressions for the contributions functions are given in table 5.2.

Contribution to Position in one Axis

The contribution of information that the observation gives to the object position along an axis is given in the corresponding diagonal element of the information matrix. The contribution functions along each axis are given in table 5.2. Note that the expressions for the contribution in position do not depend on the value of r. This is because any point of the edge contributes the same information to the position of the object.

Total Contribution to Position

The total contribution in position of one point (row P of table 5.2) only depends on the sensor precision, not on the relative location of the point with respect to the object.

Contribution to Orientation in one Axis

The values of λ for which the contributions of the point in orientation around each axis are maximal (see table 5.2) coincide with the ones that would maximize and minimize the contribution of the observation of the whole edge. An additional factor to take into account is the relative distance between the model feature and the observation, given by r. In determining the point where the contribution is minimal or maximal is it important to take into account that the edge has a limited length. Thus, the point of minimal contribution is the one calculated here if it is contained in the region occupied by the edge. Otherwise, it will be one of the extremes (the one that is closest to the point of minimal contribution). The point of maximal contribution will always be one of the extremes (the one that is further away from the point of minimal contribution).

For example, in the case where $\sigma_y = \sigma_z = \sigma$, the contribution around the x axis is:

$$\texttt{contribution}^{\psi}(p_k) = \frac{(o_x\, r - pa_x)^2 + (a_x\, r + po_x)^2}{\sigma^2}$$

Deriving with respect to r:

$$\frac{\partial}{\partial\, r}\, \texttt{contribution}^{\psi}\,(p_k) = 2\,\frac{\left(o_x^2 + a_x^2\right)\, r - pa_x\, o_x + a_x\, po_x}{\sigma}$$

The value of r which makes this expression equal to zero is:

$$r_{\text{min}} = -\frac{a_x\, po_x - pa_x\, o_x}{o_x^2 + a_x^2} \quad ; \quad \texttt{contribution}^{\psi}\,(p_k) = \frac{(po_x\, o_x + pa_x\, a_x)^2}{(o_x^2 + a_x^2)\, \sigma^2}$$

This corresponds to the point of the edge which is closest to the x axis, where the contribution of orientation around that axis is minimal (fig. 5.9).

Total Contribution to Orientation

The total contribution of the observation to the orientation of the object (row O of table 5.2) depends on the relative location of the point with respect

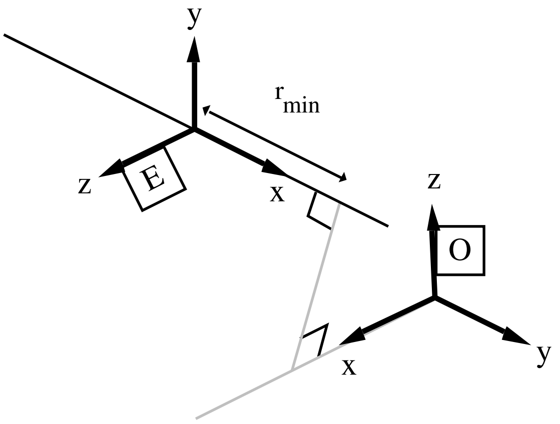


Figure 5.9: The point of an edge whose contribution to orientation around the x axis is minimal is the point closest to the axis

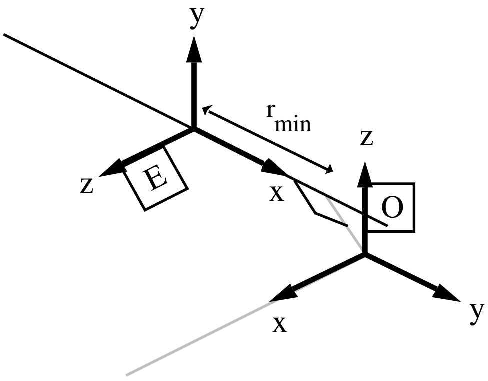


Figure 5.10: The point of the edge where the total contribution to orientation is minimal is the point closest to the origin of the reference associated to the object location

to the edge (represented by $f(r)$). It can be seen that $f(r)$ is minimal when $r = -\mathbf{p} \cdot \mathbf{n}$, that is, when the observed point coincides with the closest point of the edge to the object location (fig. 5.10).

5.4 Mobile 2D Vision

5.4.1 Computing the Contribution of a 2D Vision Edge

Consider the use the camera-in-hand sensor to obtain an observation of an edge (figure 5.11). The covariance matrix of such an observation is (see chapter 2):

$$C_{P_k} = \begin{bmatrix} \sigma_z^2 & \sigma_{x\,\psi} & 0 \\ \sigma_{x\,\psi} & \sigma_\psi^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}$$

Supposing there is no prediction error, the estimated value of the relative transformation between the observation and the prediction would be:

$$\hat{\mathbf{x}}_{P_k M_k} = (0,\ 0,\ 0,\ \lambda,\ 0,\ 0)^T$$

In this case, the binding matrix of the pairing is:

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

Thus, to calculate the information matrix F_k corresponding to such an observation, we have:

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

$$H_k = \left[\begin{matrix} o_x\, S\lambda + a_x\, C\lambda & o_y\, S\lambda + a_y\, C\lambda & o_z\, S\lambda + a_z\, C\lambda \\ 0 & 0 & 0 \end{matrix} \right.$$
$$\left. \begin{matrix} po_x\, S\lambda + pa_x\, C\lambda & po_y\, S\lambda + pa_y\, C\lambda & po_z\, S\lambda + pa_z\, C\lambda \\ o_x\, C\lambda - a_x\, S\lambda & o_y\, C\lambda - a_y\, S\lambda & o_z\, C\lambda - a_z\, S\lambda \end{matrix} \right]$$

In order to determine which value of λ maximizes the contribution of the observation, we make the following considerations:

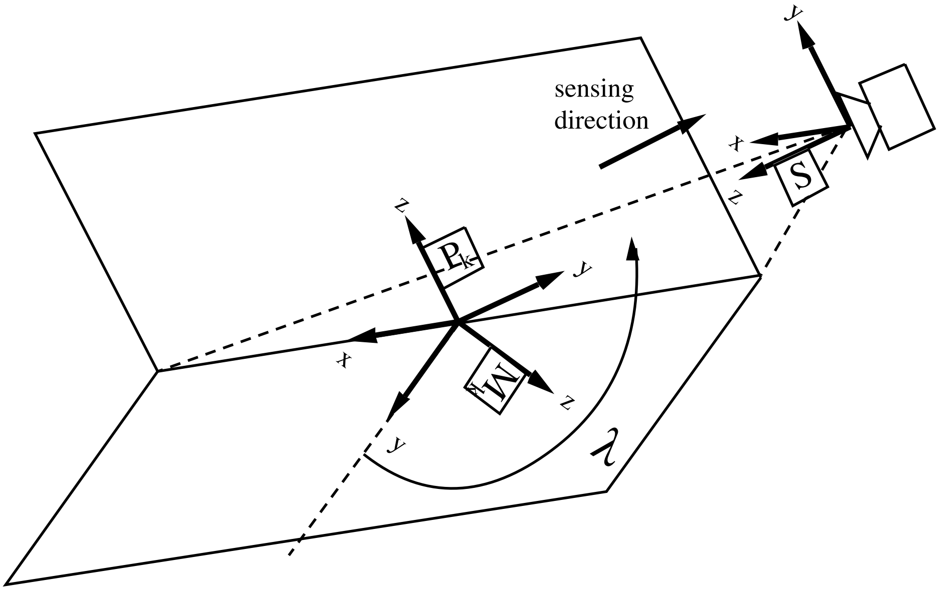


Figure 5.11: Observing an edge using 2D vision

- Let $[\lambda_m, \lambda_M]$ denote the range of visibility of the edge (considering only the local geometry of the object). We must select a sensing direction from which the edge is visible, that is $\lambda \in [\lambda_m, \lambda_M]$.
- In this case, we are free to locate the sensor in any direction around the x axis of the edge.

The optimal sensing direction is obtained by finding the value of λ where the derivate of the expression of the contribution equals zero. The expressions for the contributions on all edges are given in table 5.3.

Contribution to Position in one Axis

The contribution of the observation from the optimal sensing direction fundamentally depends on the location of the feature relative to the object location. For example, in the case of the x axis (row x of t able 5.3), when the edge is parallel to the x axis, we have that $o_x = a_x = 0$. This means that in this case the feature *cannot* contribute any information along this axis.

	$\texttt{contribution}$	$\lambda_{\mathrm{min}}; \texttt{contribution}\,(\lambda_{\mathrm{min}})$ $\lambda_{\mathrm{max}}; \texttt{contribution}\,(\lambda_{\mathrm{max}})$
x	$\frac{(o_x\, S\lambda + a_x\, C\lambda)^2}{\sigma_z^2}$	$-\tan^{-1}\left(\frac{a_x}{o_x}\right)\ ;\ 0$ $\tan^{-1}\left(\frac{o_x}{a_x}\right)\ ;\ \frac{o_x^2+a_x^2}{\sigma_z^2}$
y	$\frac{(o_y\, S\lambda + a_y\, C\lambda)^2}{\sigma_z^2}$	$-\tan^{-1}\left(\frac{a_y}{o_y}\right)\ ;\ 0$ $\tan^{-1}\left(\frac{o_y}{a_y}\right)\ ;\ \frac{o_y^2+a_y^2}{\sigma_z^2}$
z	$\frac{(o_z\, S\lambda + a_z\, C\lambda)^2}{\sigma_z^2}$	$-\tan^{-1}\left(\frac{a_z}{o_z}\right)\ ;\ 0$ $\tan^{-1}\left(\frac{o_z}{a_z}\right)\ ;\ \frac{o_z^2+a_z^2}{\sigma_z^2}$
P	$\frac{1}{\sigma_z^2}$	
ψ	$\frac{(o_x\, C\lambda - a_x\, S\lambda)^2}{\sigma_\theta^2} + \frac{(po_x\, C\lambda + pa_x\, C\lambda)^2}{\sigma_z^2}$	$\lambda\mathrm{max} \pm \frac{\pi}{2}$ $\frac{1}{2}\ \tan^{-1}\left(-\frac{2\,o_x\,a_x\,\sigma_z^2 - 2\,po_x\,pa_x\,\sigma_\theta^2}{(o_x^2-a_x^2)\sigma_z^2+(pa_x^2-po_x^2)\sigma_\theta^2}\right)$
θ	$\frac{(o_y\, C\lambda - a_y\, S\lambda)^2}{\sigma_\theta^2} + \frac{(po_y\, C\lambda + pa_y\, C\lambda)^2}{\sigma_z^2}$	$\lambda\mathrm{max} \pm \frac{\pi}{2}$ $\frac{1}{2}\ \tan^{-1}\left(-\frac{2\,o_y\,a_y\,\sigma_z^2 - 2\,po_y\,pa_y\,\sigma_\theta^2}{\left(o_y^2-a_y^2\right)\sigma_z^2+\left(pa_y^2-po_y^2\right)\sigma_\theta^2}\right)$
ϕ	$\frac{(o_z\, C\lambda - a_z\, S\lambda)^2}{\sigma_\theta^2} + \frac{(po_z\, C\lambda + pa_z\, C\lambda)^2}{\sigma_z^2}$	$\lambda\mathrm{max} \pm \frac{\pi}{2}$ $\frac{1}{2}\ \tan^{-1}\left(-\frac{2\,o_z\,a_z\,\sigma_z^2 - 2\,po_z\,pa_x\,\sigma_\theta^2}{(o_z^2-a_z^2)\sigma_z^2+(pa_z^2-po_x^2)\sigma_\theta^2}\right)$
O	$\frac{\Vert\mathbf{p}\Vert^2}{\sigma_z^2}\,(1 - f\,(\lambda)) + \frac{1}{\sigma_\theta^2}$ $f\,(\lambda) = (\cos\beta\,\sin\lambda - \cos\gamma\,\cos\lambda)^2$	$-\tan^{-1}\left(\frac{\cos\beta}{\cos\gamma}\right)\ ;\ \frac{\Vert\mathbf{p}\Vert^2}{\sigma_z^2}\,\cos^2\alpha + \frac{1}{\sigma_\theta^2}$ $\tan^{-1}\left(\frac{\cos\gamma}{\cos\beta}\right)\ ;\ \frac{\Vert\mathbf{p}\Vert^2}{\sigma_z^2} + \frac{1}{\sigma_\theta^2}$

Table 5.3: Contribution functions for an edge observed using the 2D vision sensor; σ_z and σ_θ: sensor precision; λ: sensing direction; o_x, o_y, o_z, a_x, a_y, a_z, po_x, po_y, po_z, pa_x, pa_y, and pa_z: components of the Jacobian matrix J_{M_kO}; $\mathbf{p}$: position vector of $\mathbf{x}_{OM_k}$; β and γ: angles between the position vector $\mathbf{p}$ and unit vectors $\mathbf{o}$ and $\mathbf{a}$ (see appendix A)

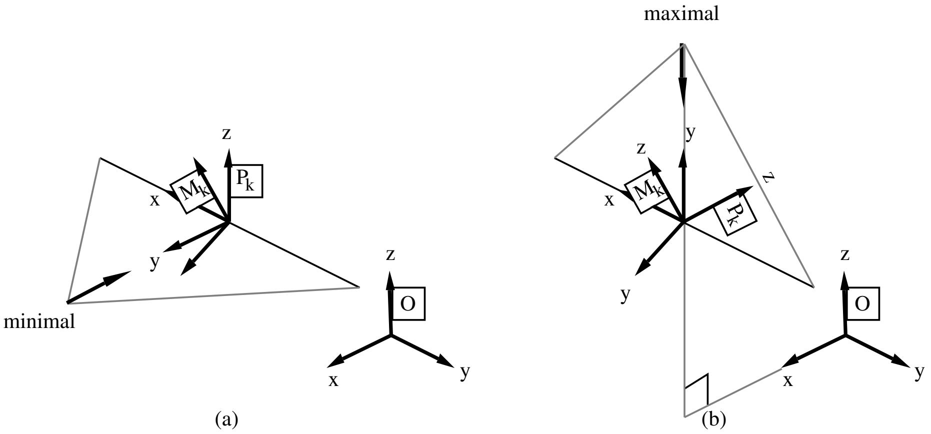


Figure 5.12: The contribution of a 2D edge to the position of the object along x axis (a) is zero when sensed from a direction parallel to the axis; (b) is *maximal* when sensed from a direction orthogonal to the axis

Total Contribution to Position

Again we can see the the total contribution of an observation to the position of the object (row P of table 5.3) is independent of the feature location and the sensing direction.

Contribution to Orientation in one Axis

The contribution of the observation to an orientation component depends fundamentally on the sensor location. For example, in the case of the ψ component (row ψ of table 5.3), the contribution is maximal when the plane formed by the edge and the focal point of the camera are parallel to the x axis. This contribution becomes zero when this plane is orthogonal to the axis (fig. 5.13).

Total Contribution to Orientation

The total contribution to orientation (row O of table 5.3) is minimal when the plane formed by the edge and the focal point of the camera is orthogonal to the position vector $\mathbf{p}$ of the edge (fig. 5.14). On the other hand, the contribution is maximal when this plane contains the origin of the object

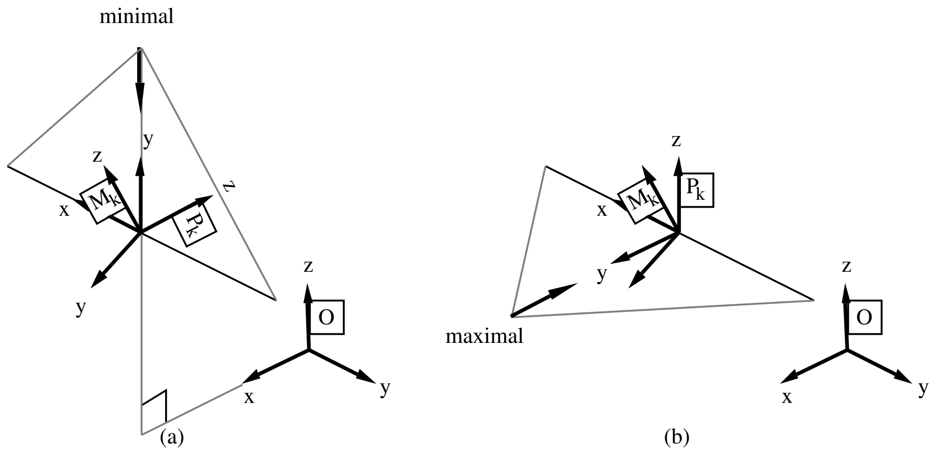


Figure 5.13: The contribution of a 2D edge to the orientation of the object around the x axis (a) is zero when sensed from a direction orthogonal to the axis; (b) is *maximal* when sensed from a direction parallel to the axis

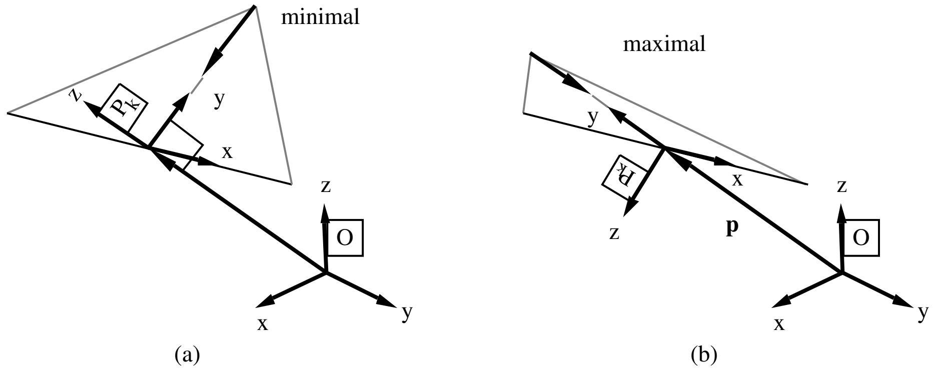


Figure 5.14: The total contribution of a 2D edge to orientation (a) is *minimal* when the plane formed by the edge and the focal point of the camera is orthogonal to the position vector **p** of the edge; (b) is *maximal* when this plane contains the origin of the object reference

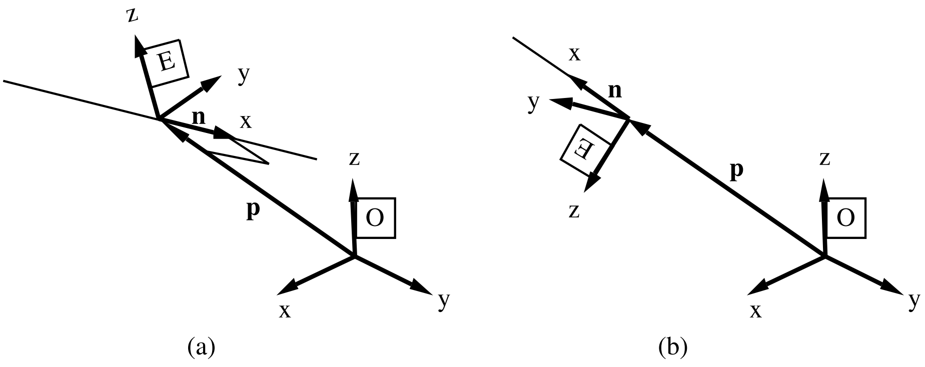


Figure 5.15: The total contribution of a 2D edge to orientation (a) is *minimal* when the edge has transversal orientation; (b) is *maximal* when the edge has radial orientation

reference, that is, when it is parallel to vector **p**. Note that the minimal solution depends on the angle α, formed by vectors **n** and **p** of the reference associated to the edge (fig. 5.15.a). When the edge has a transversal orientation, we have:

$$\cos\alpha = 0 \;\; ; \;\; \texttt{contribution}^{O}(p_k) = \frac{1}{\sigma_\theta^2}$$

When the edge has radial orientation ($\mathbf{p}||\mathbf{n}$) (fig. 5.15.b), the given solution is not valid because $\cos\alpha = \pm 1$ implies $\cos\beta = \cos\gamma = 0$ (see appendix A). In this case the relative orientation of the camera has no influence in the total contribution of the feature to the orientation of the object, which always is:

$$\texttt{contribution}^{O}(p_k) = \frac{||\mathbf{p}||^2}{\sigma_z^2} + \frac{1}{\sigma_\theta^2}$$

In addition to the fact the maximal and minimal sensing directions are orthogonal, in this case we can also see that the sensor directions where the contribution to position along one axis are maximal, coincide with the sensing directions where the contribution to orientation around the same axis is minimal.

5.5 The Potential Contribution of a Model Feature

In refining the estimated location of an object, it is necessary to determine is how much location information each feature can contribute to each component of the object location. This information will help us in selecting the model feature whose location should be verified in order to reduce the uncertainty in the component of the location vector that we have as goal.

In the previous section we have seen that there are three fundamental factors that determine the contribution of location information of a feature: *sensor precision*, *feature dimensions* and *feature location*. In order to compare the potential benefit of different features on a fair basis, we will make use again of the *prototypical sensor* defined in section 4.3.4. For this purpose we will consider that this prototypical sensor is capable of observing points, whose covariance is given by:

$$C_P = \mathrm{diag}(\sigma^2,\ \sigma^2,\ \sigma^2)$$

Assuming this, we can calculate the potential contribution of different model features. Continuing with our edge examples, we have that according to table 5.1, the contributions of an edge to each component of the object location vector are given by:

$$\begin{aligned}
\texttt{contribution}^x(m_k) &= \frac{n\,(o_x^2 + a_x^2)}{\sigma^2} \\
\texttt{contribution}^y(m_k) &= \frac{n\left(o_y^2 + a_y^2\right)}{\sigma^2} \\
\texttt{contribution}^z(m_k) &= \frac{n\,(o_z^2 + a_z^2)}{\sigma^2} \\
\texttt{contribution}^\psi(m_k) &= \frac{s^2 n^3\left(\frac{o_x^2}{12} + \frac{a_x^2}{12}\right) + n\,(po_x^2 + pa_x^2)}{\sigma^2} \\
\texttt{contribution}^\theta(m_k) &= \frac{s^2 n^3\left(\frac{o_y^2}{12} + \frac{a_y^2}{12}\right) + n\left(po_y^2 + pa_y^2\right)}{\sigma^2} \\
\texttt{contribution}^\phi(m_k) &= \frac{s^2 n^3\left(\frac{o_z^2}{12} + \frac{a_z^2}{12}\right) + n\,(po_z^2 + pa_z^2)}{\sigma^2}
\end{aligned}$$

We can use these functions to determine the potential contribution of location information of a set of model features.

R	$\mathbf{x}_{OR}$	len	x	y	z	ψ	θ	ϕ
A	$(0, 100, 0, 0, 0, 0)^T$	200	0	5	5	50000	26042	26042
B	$(0, 100, -100, 0, 0, 0)^T$	200	0	5	5	100000	26042	26042
C	$(100, 0, -100, 0, 0, \pi/2)^T$	200	5	0	5	26042	100000	26042
D	$(100, -50, 100, 0, 0, \pi/2)^T$	100	3	0	3	13125	60000	13125
E	$(-100, -50, 100, 0, 0, \pi/2)^T$	100	3	0	3	13125	60000	13125
F	$(-100, 50, 50, 0, -\pi/4, -\pi/2)^T$	142	3	1.5	1.5	5625	25313	25313

Table 5.4: Potential contribution for the labeled edges in each of the components of the object location vector

Example 5.1: *Potential contribution of model edges*

Consider the object shown in figure 5.16. We can use the results above to calculate the potential contribution of each of the edges (labeled A, B, C, D, E, and F). The results are shown in table 5.4. For the calculation of this values we have used $s = 50mm$, and have eliminated σ from the denominator. From the edge length l and s we can calculate the value of n as $n = l/s + 1$.

Several facts can be drawn from this table:

- Edges A and B cannot contribute location information in the x component. Edge C is the one that contributes more information, because it is the longest one that is appropriately aligned.
- Edge A and B are the ones that can contribute more information in the y component, because of their orientation, but edge F can also contribute information.
- All edges can contribute information to the z component because none is aligned with the z axis of the object location.
- The contribution to orientation depends on the distance between the edge and the object reference, and on its relative orientation. For this reason, even though edges A, B, and C have the same length, edge B contributes more to ψ, and edge C contributes more to θ.

◇

It must be taken into account that in this *a priori* analysis there are several factors that are not taken into account. The first is the local geometry of the feature. It can happen that an edge may contribute less information than its potential if the relative angle between its adjacent faces precludes it. Additionally, the object location may make the edge inaccessible to sensors.

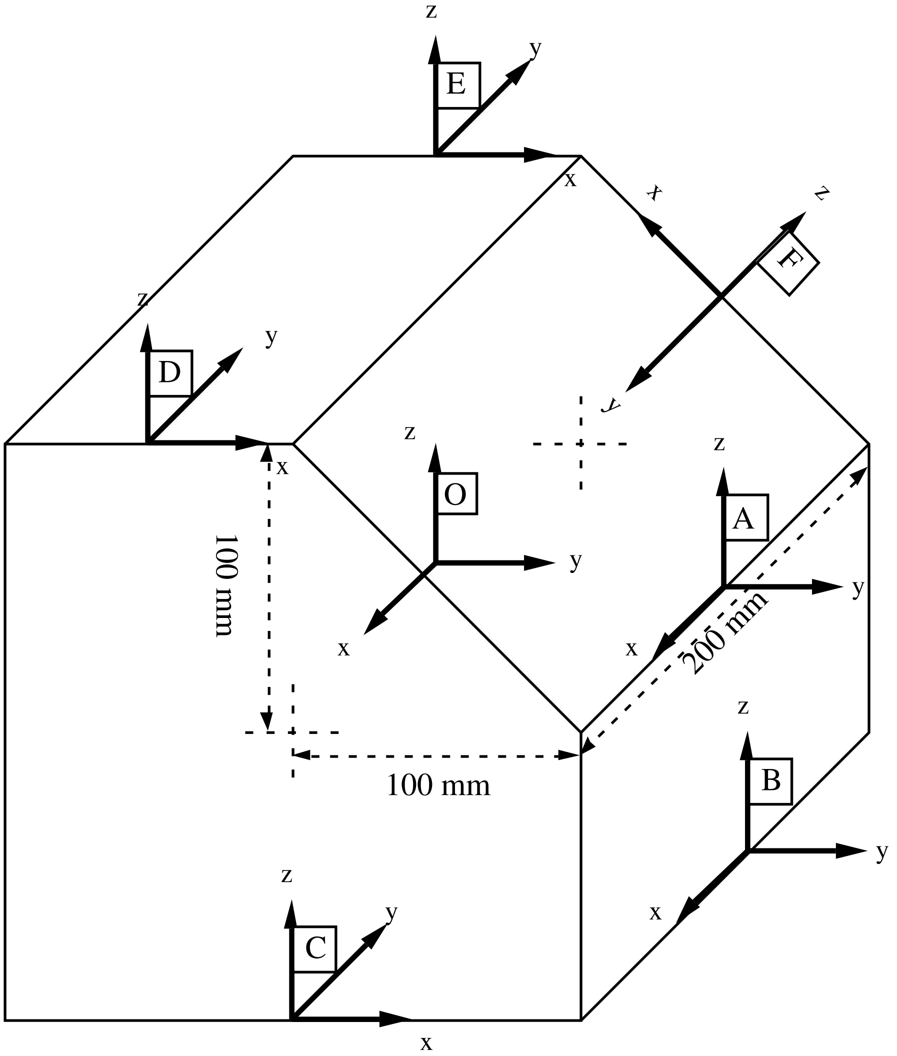


Figure 5.16: Object model

The last factor is related to the capacity of the observation in reducing the *coupling* between the position and orientation terms of a hypothesis. We can see from the table that edge D and E give exactly the same potential contributions in all components, but one of the two may lie in an unobserved region of the object, and for this reason its observation would balance the hypothesis better.

Given that an object may potentially have a considerable number of features, we will use these potential contribution tables to make an initial selection of the model feature whose location in the scene we should predict. Limiting the number of features for which we must do further analysis, in which the factors mentioned above are taken into account, will allow us to make an adequate selection at a reasonable cost.

5.6 Sensor and Sensor Location Selection

Consider an object-location hypothesis, whose estimated location is given by $\mathbf{L}_{WO} = (\hat{\mathbf{x}}_{WO}, \hat{\mathbf{d}}_O, C_O)$. In verifying this object-location hypothesis and refining its estimated location, we proceed in the following way:

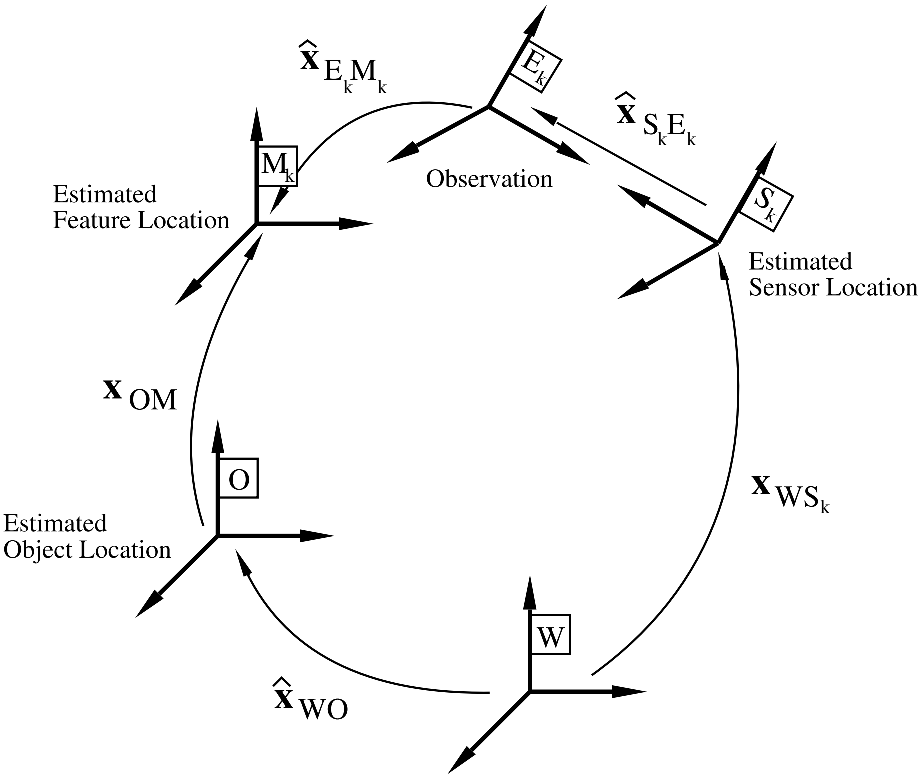
1. Select the most uncertain component of the location vector as the goal for uncertainty reduction. This can be done by determining the greatest diagonal element of the normalized covariance matrix $\overline{C}_O$. Let α be such component.

2. Using the potential contribution table of the object, select a model feature e_k whose observation may contribute more location information to the goal α.

3. Given the predicted location of the feature in the scene, $\hat{\mathbf{x}}_{WM_k}$, consider a set of sensors $\{S_1, \cdots, S_n\}$ to verify the location of this feature. Compute the maximal possible contribution for each sensor, and choose the one that gives the greatest maximal contribution. Let S_m be such sensor.

4. Let $\hat{\mathbf{x}}_{E_m M_k}$ be the relative location between the observation and model feature according to this sensor, where its contribution is maximal. Compute the sensor location as follows (fig. 5.17):

$$\hat{\mathbf{x}}_{WS_m} = \hat{\mathbf{x}}_{WO} \oplus \mathbf{x}_{OM_k} \ominus \hat{\mathbf{x}}_{E_m M_k} \ominus \hat{\mathbf{x}}_{S_m E_m} \qquad (5.3)$$



$\hat{\mathbf{x}}_{E_k M_k}$
E_k
$\hat{\mathbf{x}}_{S_k E_k}$
Observation
M_k
Estimated Feature Location
S_k
Estimated Sensor Location
$\mathbf{x}_{OM}$
O
$\mathbf{x}_{WS_k}$
Estimated Object Location
W
$\hat{\mathbf{x}}_{WO}$

Figure 5.17: References involved in calculating the sensor location

where the location vector $\hat{\mathbf{x}}_{WO}$ represents estimated object location, $\mathbf{x}_{OM_k}$ is the model feature location relative to the object location, $\hat{\mathbf{x}}_{E_m M_k}$ represents the expected transformation between the observation and the model feature where its contribution is maximal, and $\hat{\mathbf{x}}_{S_m E_m}$ the relative location between the sensor and the observation.

Example 5.2: *Sensor selection to observe an edge*
Consider the object in figure 5.18. Suppose the system chooses to verify the presence of the edge whose reference is A. The relative location of A with respect to the object is $\mathbf{x}_{OA} = (0,\ 100,\ 0,\ 0,\ 0,\ 0)^T$. Its range of visibility is $[-2.35, 1.57]$. In this case, the two admissible orientation ranges for the proximity sensor are $\lambda_1 = 0 \pm \epsilon$ and $\lambda_2 = .78 \pm \epsilon$. The contribution functions for the y, z, θ, and ϕ components of the object location vector are given in figure 5.19 (these functions, from tables 5.1 and 5.3, are plotted over the range of visibility of the feature).

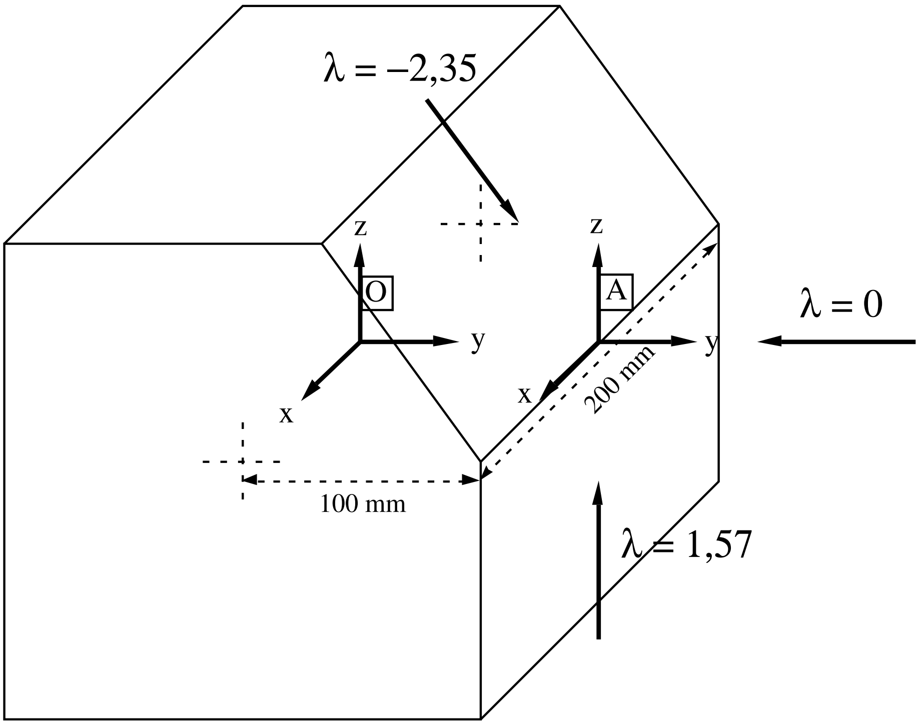


Figure 5.18: Viewing angles from observing edge A

There are two general comments that we can make regarding these contribution graphs:

- We can notice that, in general, it seems that the proximity sensor contributes more information to *position* than the vision sensor. This is due to the fact that the position errors of proximity points due to the location errors of the robot are independent, and thus tend to compensate each other. This is not the case in 2D vision, where the position of all pixels is affected by the same robot location error. In the case of *orientation* this is less evident, because 2D vision acquires many more points (pixels) than proximity (we have considered 200 pixels for 2D vision, and 4 proximity points).

- It seems that the best positions for one sensor seem to be the worst for the other. This happens because proximity measures distance along the y axes of the resulting observation (fig. 5.2), while 2D vision perceives position along the z axis of the resulting observation (fig. 5.11).

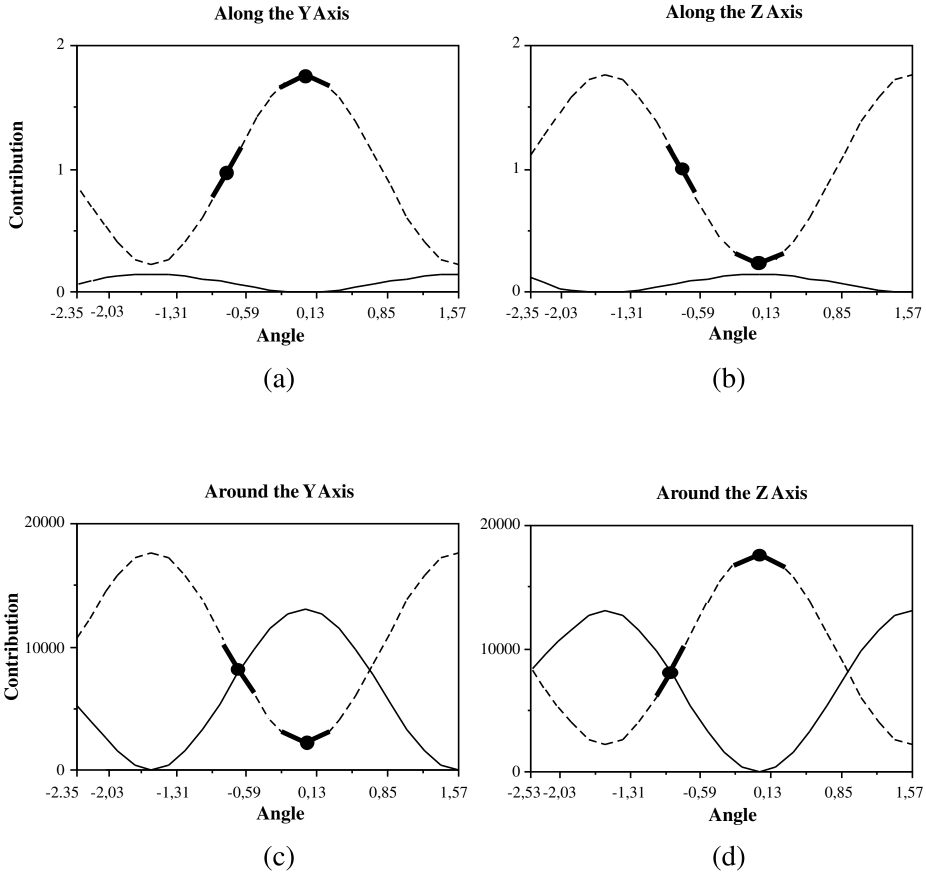


Figure 5.19: Contribution of an edge on the object location, as a function of the observation angle. (2D-Vision: — , Proximity: - - - , Admissible sensor locations: •)

Let us now analyze which sensor should be used for different perception tasks:

1. Suppose our goal is to reduce uncertainty along the y axis. From fig. 5.19.a we can see that the proximity sensor is clearly more suitable. Note that the sensor orientation where the contribution along the y axis is maximal, is the same sensor orientation where the contribution along the z axis is minimal (fig. 5.19.b).
2. If our goal is to reduce uncertainty along the z axis (fig. 5.19.b) proximity is still best, but its contribution is not as much as in the preceding case, because of sensor positioning constraints. The contribution of an observation along the z using proximity is smaller due to the fact that it can only be positioned in directions nearly normal to the sensed surface.
3. If the goal of the perception task is to reduce uncertainty in the θ component (around the y axis, fig. 5.19.c), 2D vision can contribute more information, while if the goal is to reduce uncertainty in the ϕ component (around the z axis, fig. 5.19.d), proximity contributes more information.

◇

5.7 Conclusions

This chapter is concerned with the characterization of the location information contributed by a sensing operation, and with the determination of the location of the sensor so that this contribution is maximized. The improvement of the estimation of an object location due to the integration of an observation is modeled as a function of the involved geometric element, the sensor characteristics and its location relative to the observed feature. This characterization is very useful for the strategic planning of perception tasks, by allowing to compare the potential benefit of a sensing operation to its estimated cost. There are three general conclusions that we can draw from this work:

- For a given sensor, the selection of its location greatly depends on the goal of the perception task. Sensing directions which are maximal for some goal, are minimal for some other.
- Sensor location constraints can have considerable influence on its capacity to reduce uncertainty, especially in the case of proximity sensors.
- The object geometry also influences the usefulness of a feature in reducing location uncertainty. The range of visibility of the feature may limit sensor positioning.

In general, the usefulness and reliability of a sensing operation are inversely proportional to its cost. More discriminant features are in general larger and more complex, and thus more costly to observe. With respect to reliability, those sensing operations that can contribute more information are related to features and areas of the scene of which we are more uncertain. We have calculated the estimated location of the sensor where the contribution of its observation to the object location is maximal. Calculating the uncertainty of this estimation, which depends on the precision of the hypothesized feature location, allows us to bound the region where the sensor must search for the feature. The size of this region can give an idea of how costly this sensor operation will be. Future efforts will be directed to estimate the trade-off between the cost of a sensing operation and its potential benefit.

In this work, the selection of the model feature and of the sensor and sensor location used to verify its presence is done considering each object-location hypothesis in a separate way. In this sense, the utility of the perception task is limited to determining whether the involved object-location hypothesis is correct or not, and to refine the estimated location of the object, in case the hypothesis is correct. Future work may be directed to the analysis of perception tasks allowing to disambiguate between several competing hypotheses.

Chapter 6

Identifying while Locating

Summary

In this chapter we study the relationship between the two fundamental aspects of recognition: identification *and* localization*. We analyze the advantages of the* identifying while locating *recognition scheme in which identification and localization are performed simultaneously. We show that the identification process benefits from having an estimation of the object location because constraint analysis can be less costly and more discriminant. The probability of generating an incorrect interpretation of the sensorial data is reduced, and the availability of* a priori *information for sensors allows the use of less costly and more reliable perception processes. Within this recognition scheme, we propose the use of several strategies for the generation and verification of hypotheses that can help reduce the complexity of recognition [Neira 93c].*

6.1 Introduction

Identification and localization may be separate processes. One may accumulate sufficient evidence on the identity of an object without estimating its location. One can also have a small set of observations that allow to have an estimation of the object location, and yet they may be insufficient to assert the identity of the object. Consider the object in figure 6.1.a. Intuitively, the available set of observations (fig. 6.1.b) strongly suggests that they come from the specified object. Nevertheless, it is not possible to determine the object location. In terms of precision and relevance, this object location hypothesis has high relevance and zero precision (without considering extension constraints, there is infinite location uncertainty in one direction).

The relationship between the processes that must attain identification and localization has given place to different recognition schemes. Some approaches follow the *Identifying before Locating* scheme, in which identification is performed prior to localization. The work of Grimson [Grimson 84, Grimson 87, Grimson 90a] follows this approach. Grimson states the recognition problem as a search in an interpretation tree, whose nodes are constituted by observation-model pairings, providing partial interpretations of a set of sensor observations, and thus, representing the solution space. In order to prune the search and make the process more efficient, he uses location independent geometric constraints–simple and fast tests—as the fundamental source of information to validate the consistency of an interpretation. An estimation of the object location is not required to perform these tests.

Alternative approaches are based on the *Identifying while Locating* scheme, where identification and localization are executed simultaneously. In it, object location hypotheses are generated with the minimum number of observations needed to estimate object location. A hypothesis is then verified by a process in which the location of new features is predicted, and its presence is verified among the available data or by a sensor. This scheme is used in [Bolles 82] to identify and locate two dimensional objects. The approach of Bolles is based on concentrating on the observation of small clusters of focus features, that the system considers more relevant for the identification of the object. Verifying the presence of these focus features requires estimating the object location and updating it with the inclusion of each matched observation. The 3DPO System [Bolles 86] extends this approach to the three dimensional case.

This recognition scheme is also used in [Faugeras 86]. In this work, recognition starts with a hypothesis generation process, in which the system selects the smallest number of observations that allow it to determine the object location. Object location hypothesis are generated by searching for compatible features in the database of object models. A hypothesis verification process takes place, in which the location of new features in the scene is predicted and their presence verified. Emphasis is made on the use of the *rigidity* constraint in the recognition process as early as possible, since it is the only constraint that guarantees global consistency. In [Lozano-Pérez 87], Lozano-Pérez et al. implement this recognition scheme for the robot system Handey, which locates objects from a pile, from a set of linear segments of a depth map acquired using a light-stripping triangulation sensor.

One of the fundamental steps of recognition consists in obtaining an initial estimation of the object location from a set of features. A first problem consists in deciding whether a set of observations are *independent* (in an algebraic sense), that is, allow to completely determine location of the object. A pair of geometric features are independent if they have no common symmetries of continuous motion. Thomas [Thomas 88a] and Tardós [Tardós 91] give a catalog of the conditions under which a pair of features are independent. In these works, the uncertainty associated to the location of sensorial observations is not considered. A second problem we face is calculating the object location, and its precision. The majority of works solve this problem in a feature specific way. However, no general procedure has been described.

The purpose of this chapter is to show that simultaneous identification and localization may result in a more efficient recognition scheme. In section 6.2 we analyze the complexity of both recognition schemes from the constraint validation point of view. In section 6.3 we derive a general procedure to determine whether a set of uncertain observations allow to determine the location of the object and to calculate this location. This analysis will lead us to propose heuristics for the generation and verification of hypotheses, in sections 6.4 and 6.5.

6.2 Constraint Based Recognition

The idea behind the *identifying before locating* scheme is to avoid the computational cost of estimating the object location for a hypothesis that may be later dismissed. However, an estimation of the object location constitutes a very important source of information for the following reasons:

1. *The number of constraints to verify is smaller:* according to the *identifying before locating* scheme, given a hypothesis with n observation-

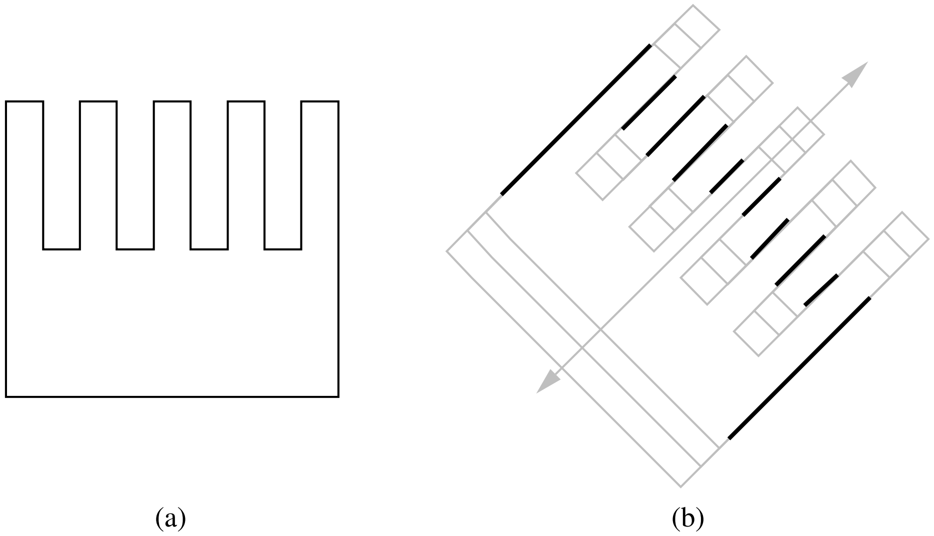


Figure 6.1: Identifying without locating

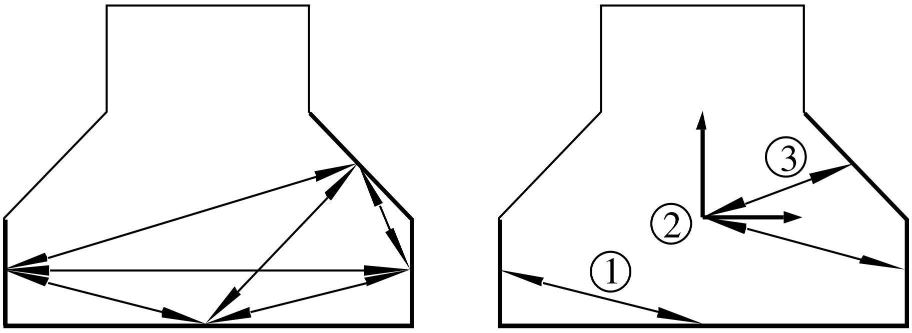
model pairings, the inclusion of another pairing requires the validation of binary constraints between the new pairing and each of the n existing pairings in the hypothesis. This means that the number of binary constraints applied to validate a hypothesis with n pairings is:

$$\frac{n(n-1)}{2}$$

On the other hand, in the *identifying while locating* approach, having located the object, the only constraint that must be validated for each new pairing is rigidity: the observation and predicted model feature locations must coincide. Thus in this case the number of constraints that would have been validated for the hypothesis is:

$$n - 1$$

In the example of figure 6.2.a, the *identifying before locating* approach is used, and thus, each included edge is validated with all the preceding edges. Figure 6.2.b represents the use of the *identifying while locating* approach, where only the location independent geometric relations for



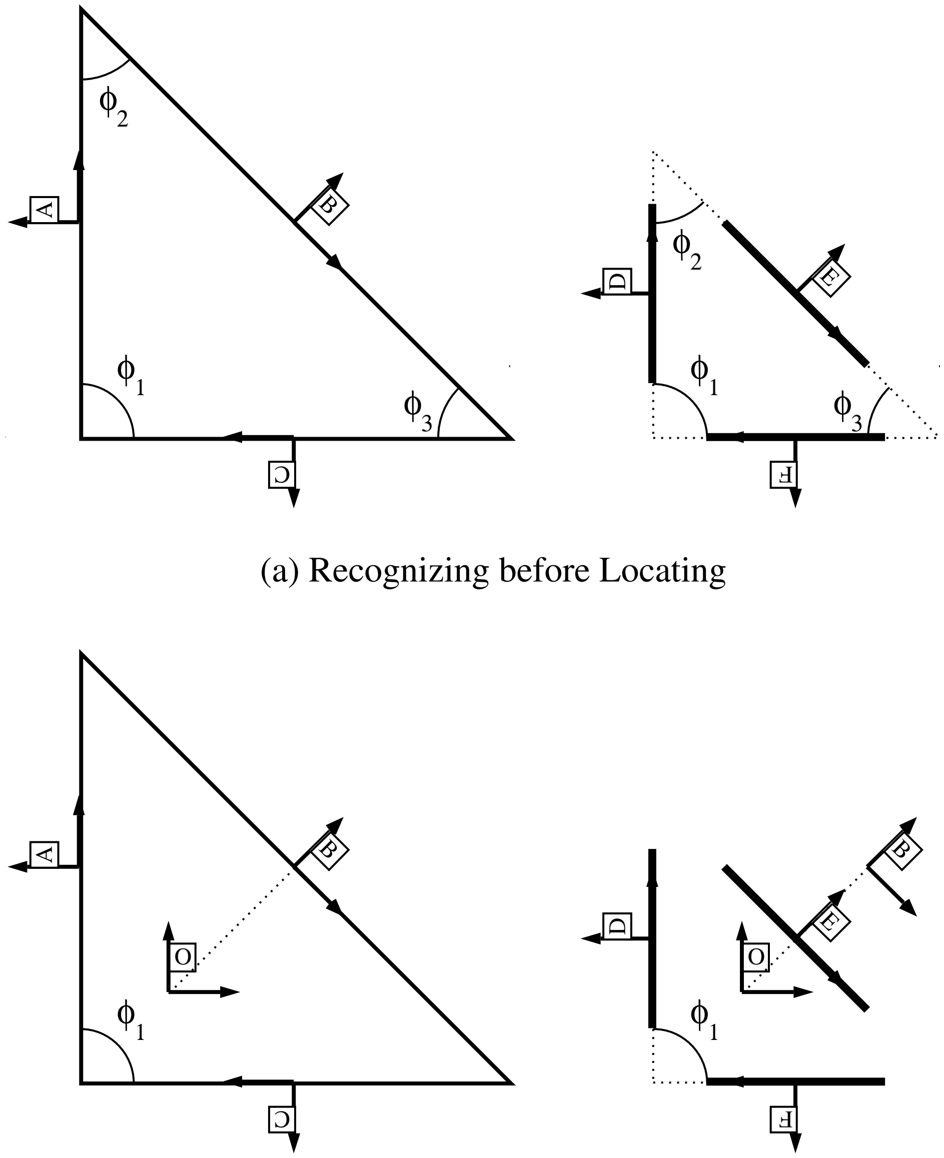
(a) Recognizing before Locating (b) Recognizing while Locating

Figure 6.2: Complexity of constraint validation

the first two edges are validated (step 1); these two edges are used to determine the object location (step 2) and each additional edge is validated using the rigidity constraint (step 3).

2. *Location dependent constraints are tighter:* a set of pairings may satisfy all location independent constraints, and yet the interpretation may not be globally consistent. This implies that in the *identifying before locating* scheme, many inconsistent hypotheses may survive until the object location is estimated. This is because location independent constraints only assure *local* consistency, while the availability of an estimation of the object location allows the validation of location dependent constraints, which assure *global* consistency. In the example of figure 6.3.a (from [Grimson 90a]), we can see that the pairings $\{(A, D), (B, E), (C, F)\}$ satisfy all constraints even though the interpretation is not globally consistent (there is no object location where the location of all the observed features coincides with the estimated location of their corresponding model feature). The *identifying while locating* approach has the advantage that it allows to validate the tightest possible geometric constraint, which is rigidity. In figure 6.3.b we can see that if the object is located using A and C, and the location of B in the scene is predicted, it does not coincide with the location of E, and thus the interpretation would be discarded.

Thus, from the constraint validation point of view, the complexity of the *identifying while locating* approach is lower than that of the *identifying before*



(a) Recognizing before Locating

(b) Recognizing while Locating

Figure 6.3: Constraint tightness of compound features

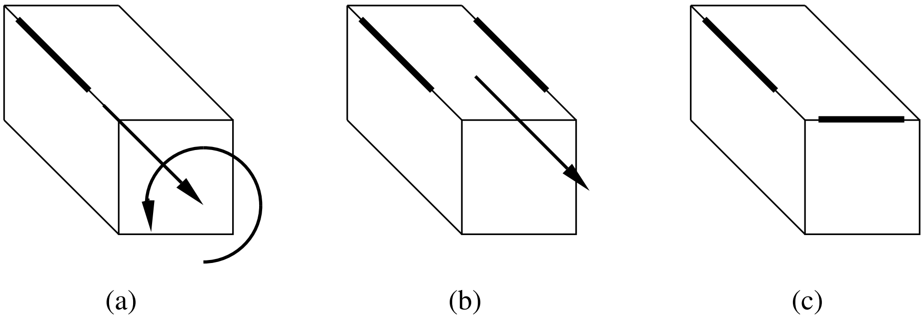


Figure 6.4: Common symmetries of edges

locating approach. This highlights the importance of having an estimation of the object location as soon as possible. In the next section we will propose a general method to determine whether a set of uncertain observations allows to determine the location of the object, and to compute this location.

6.3 Determining the Object Location

Each feature bounds some degrees of freedom in the location of the object. Which d.o.f. are bound depends on the symmetries of the feature. For example, an edge determines two d.o.f. in position and two in orientation (fig. 6.4.a). Thus, with only one edge, one d.o.f. in position and one in orientation remain unbound. At a first glance, two edges should suffice to determine the location of an object. Nevertheless, if the two edges have *common symmetries*, as in the case of parallel edges, there would still remain some unbound d.o.f. on the location of the object (fig. 6.4.b). To completely determine the location of the object, a set of features must *not* have common symmetries of continuous motion, i.e., the intersection of the symmetries of the features must be equal to the identity transformation $\mathbb{1}$ (fig. 6.4.c). Catalogs of the conditions under which two features have no common symmetries can be found in [Thomas 88a] and [Tardós 91].

In this section we describe such a catalog in an appropriate way to take into account the uncertainty related to the location of the involved geometric elements, and to obtain an estimation of the object location.

6.3.1 Common Symmetries of a Set of Observations

Let $\{E_1, E_2, \ldots, E_n\}$ represent the location of a set of n geometric elements with symmetries $\{\mathcal{S}_{E_1}, \mathcal{S}_{E_2}, \ldots, \mathcal{S}_{E_n}\}$, respectively. The symmetries of each feature are expressed with respect to the reference associated to the feature. In order to calculate the intersection, we must express each set of symmetries with respect to a common reference A. Thus, the common symmetries of the set of features with respect to A can be calculated as:

$$\begin{aligned} {}^{A}\mathcal{S}_E &= \bigcap_{i=1}^{n} {}^{A}\mathcal{S}_{E_i} \\ &= \bigcap_{i=1}^{n} t_{AE_i} \cdot \mathcal{S}_{E_i} \cdot t_{AE_i}^{-1} \qquad (6.1) \end{aligned}$$

Let us for the moment concentrate on the case when $n = 2$. The first problem we must deal with is the selection of an appropriate reference for calculating the common symmetries of the geometric features. In section 3.2.1 we have seen that given two features whose location is represented by E_1 and E_2, we can find references $\bar{E}_1$ and $\bar{E}_2$ such that $t_{\bar{E}_1\bar{E}_2}$ contains the minimum number of translations and rotations. In general, $\bar{E}_1$ and $\bar{E}_2$ are uniquely determined. Taking advantage of this, we will choose to calculate the set of common symmetries with respect to $\bar{E}_1$. In case the resulting set of symmetries is equal to $\mathbb{1}$, $\bar{E}_1$ can be used to calculate the location of the object, as we will see later. Thus, according to equation (6.1) we have:

$$ {}^{\bar{E}_1}\mathcal{S}_E = \mathcal{S}_{E_1} \cap \left(t_{\bar{E}_1\bar{E}_2} \cdot \mathcal{S}_{E_2} \cdot t_{\bar{E}_2\bar{E}_1} \right) \qquad (6.2)$$

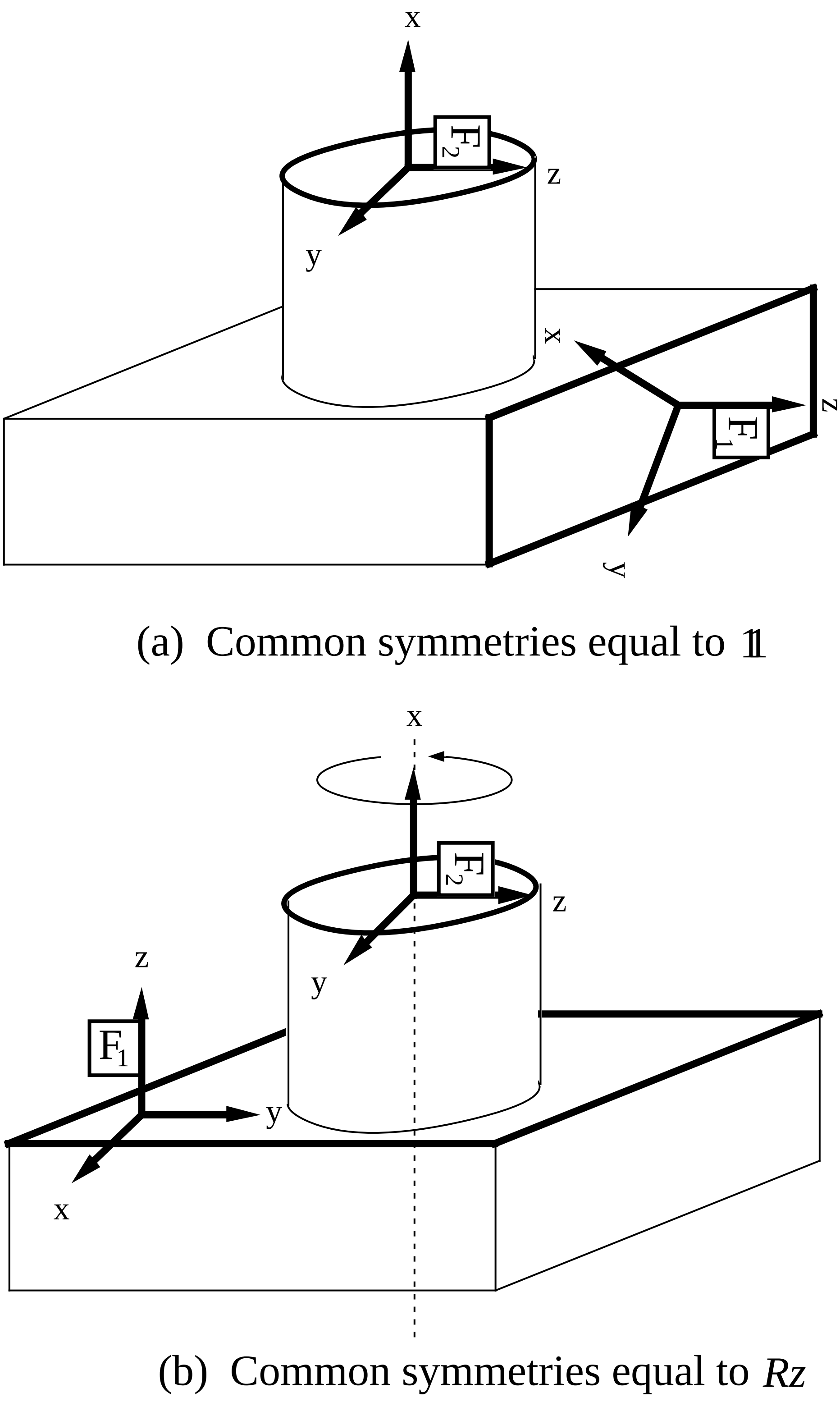
In order to be able to uniquely determine the object location, this set of symmetries must be equal to the identity. So we can deduce the values of $t_{\bar{E}_1\bar{E}_2}$ that guarantee this condition. Let us study an example.

Example 6.1: *Common symmetries of a circle and a planar surface*
Let E_1 and E_2 represent the location of a planar surface and a circular arc, respectively (fig. 6.5). In this case we have:

$$\mathcal{S}_{E_1} = T_{xy} \cdot R_z \quad ; \quad \mathcal{S}_{E_2} = R_x$$

From table 3.1 we have that the general form of $t_{\bar{E}_1\bar{E}_2}$ is $t_z(\bar{z}) \cdot r_y(\bar{\theta})$. Thus, according to (6.2), set of common symmetries of the features with respect to $\bar{E}_1$ is:

$$\begin{aligned} {}^{\bar{E}_1}\mathcal{S}_E &= (T_{xy} \cdot R_z) \\ &\cap \ \left(t_z(\bar{z}) \cdot r_y(\bar{\theta}) \cdot R_x \cdot r_y(-\bar{\theta}) \cdot t_z(-\bar{z}) \right) \end{aligned}$$



(a) Common symmetries equal to $\mathbb{1}$

(b) Common symmetries equal to Rz

Figure 6.5: Symmetries for a circle and a planar surface

Any transformation t belonging to the intersection must satisfy:

$$\begin{aligned} t &= t_x(x_a) \cdot t_y(y_a) \cdot r_z(\phi_a) \\ t &= t_z(\bar{z}) \cdot r_y(\bar{\theta}) \cdot r_x(\psi_b) \cdot r_y(-\bar{\theta}) \cdot t_z(-\bar{z}) \end{aligned}$$

Computing the corresponding location vectors and equating their terms we obtain the following six equations:

$$x_a = \bar{z}\,\cos\bar{\theta}\sin\bar{\theta}\,(1-\cos\psi_b) \tag{6.3}$$

$$y_a = \bar{z}\,\cos\bar{\theta}\sin\psi_b \tag{6.4}$$

$$0 = \bar{z}\,\cos^2\bar{\theta}\,(1-\cos\psi_b) \tag{6.5}$$

$$0 = \tan^{-1}\left(\frac{\cos\bar{\theta}\sin\psi_b}{\sin^2\bar{\theta}+\cos^2\bar{\theta}\cos\psi_b}\right) \tag{6.6}$$

$$0 = \tan^{-1}\left(\frac{\cos\bar{\theta}\sin\bar{\theta}\,(\cos\psi_b-1)}{\sqrt{\sin^2\bar{\theta}\cos\psi_b+\cos^2\bar{\theta})^2+\sin^2\bar{\theta}\sin^2\psi_b}}\right) \tag{6.7}$$

$$\phi_a = -\tan^{-1}\left(\frac{\sin\bar{\theta}\sin\psi_b}{\cos^2\bar{\theta}+\sin^2\bar{\theta}\cos\psi_b}\right) \tag{6.8}$$

From (6.6) and (6.7) we can deduce that $\psi_b = 0 \vee \bar{\theta} = \pm\pi/2$, giving two solutions:

- If $\psi_b = 0$ then the set of equations becomes:

$$\begin{aligned} x_a &= 0 \\ y_a &= 0 \\ 0 &= 0 \\ 0 &= 0 \\ 0 &= 0 \\ \phi_a &= 0 \end{aligned}$$

  This means that in the general case, the only transformation that belongs to the intersection is the identity $\mathbb{1}$.

- If $\bar{\theta} = \pm\pi/2$, then we have:

$$\begin{aligned} x_a &= 0 \\ y_a &= 0 \\ 0 &= 0 \end{aligned}$$

$$\begin{aligned} 0 &= 0 \\ 0 &= 0 \\ \phi_a &= \mp\psi_b \end{aligned}$$

This case corresponds to the situation where the circle and the plane are parallel (fig. 6.5 (b)). In this case equation 6.1 becomes:

$$\begin{aligned} {}^{\bar{E}_1}\mathcal{S}_E &= (T_{xy} \cdot R_z) \\ &\cap \ (t_z(\bar{z}) \cdot r_y(\pi/2) \cdot R_x \cdot r_y(-\pi/2) \cdot t_z(-\bar{z})) \\ &= (T_{xy} \cdot R_z) \cap (t_z(\bar{z}) \cdot R_z \cdot t_z(-\bar{z})) \\ &= (T_{xy} \cdot R_z) \cap R_z \\ &= R_z \end{aligned}$$

Thus, a planar surface and a circular arc can be used to establish the location of the object when $\bar{\theta} \neq \pm\pi/2$. ◇

The solutions of all the combinations of subgroups of symmetries can be derived in a similar way, and they are shown in table 6.1.

In most cases, two features are sufficient to determine the location of an object. But in some cases more than two observations are needed. The calculation of the set of common symmetries of more than two features can be easily done in an iterative way:

1. A pair of features is selected and the intersection of their symmetries is calculated.

2. If it is not equal to $\mathbb{1}$, the pair of features is grouped into a *compound feature*. This new feature will have an associated reference equal to the aligning reference $\bar{E}_1$ of the first feature that composes it, and its set of symmetries will be the common symmetries of the two features ${}^{\bar{E}}\mathcal{S}_E$.

3. Another feature is selected to calculate its common symmetries with the compound feature. This process continues until you end up with a reference that expresses the location of the set of features, and the set of common symmetries of the features expressed in that reference is $\mathbb{1}$.

However, it is important to note that the location of $\bar{E}_1$ resulting from this process depends on the order in which we consider the features. In

$\mathcal{S}_A$	$\mathcal{S}_B$	Case of $t_{\bar{A}\bar{B}}$	$\mathcal{S}_A \cap \mathcal{S}_B$
R_x	R_x	$\overline{y}=\overline{z}=0,\ \overline{\phi}=0,\ \pi$	R_x
		$otherwise$	$\mathbb{1}$
R_x	R_{xyz}	$\overline{z}=0$	R_x
		$otherwise$	$\mathbb{1}$
T_x	T_x	$\overline{\theta}=0,\ \overline{\phi}=0,\ \pi$	T_x
		$otherwise$	$\mathbb{1}$
T_x	R_x		$\mathbb{1}$
T_x	$T_x\,R_x$	$\overline{\theta}=0,\ \overline{\phi}=0,\ \pi$	T_x
		$otherwise$	$\mathbb{1}$
T_x	R_{xyz}		$\mathbb{1}$
$T_x\,R_x$	R_x	$\overline{y}=\overline{z}=0,\ \overline{\phi}=0,\ \pi$	R_x
		$otherwise$	$\mathbb{1}$
$T_x\,R_x$	$T_x\,R_x$	$\overline{z}=0,\ \overline{\phi}=0,\ \pi$	$T_x\,R_x$
		$\overline{\phi}=0,\ \pi$	T_x
		$otherwise$	$\mathbb{1}$
$T_x\,R_x$	R_{xyz}	$\overline{z}=0$	R_x
		$otherwise$	$\mathbb{1}$
$T_{xy}\,R_z$	R_x	$\overline{\theta}=\pm\pi/2$	R_z
		$otherwise$	$\mathbb{1}$
$T_{xy}\,R_z$	R_{xyz}		R_z
$T_{xy}\,R_z$	T_x	$\overline{\theta}=0$	T_x
		$otherwise$	$\mathbb{1}$
$T_{xy}\,R_z$	$T_x\,R_x$	$\overline{\theta}=0$	T_x
		$\overline{\theta}=\pm\pi/2$	R_z
		$otherwise$	$\mathbb{1}$
$T_{xy}\,R_z$	$T_{xy}\,R_z$	$\overline{\phi}=0,\ \pi$	$T_{xy}\,R_z$
		$otherwise$	T_y
R_{xyz}	R_{xyz}	$\overline{z}=0$	R_{xyz}
		$otherwise$	R_z

Table 6.1: Intersection of the symmetries of two geometric elements, expressed in $\overline{A}$.

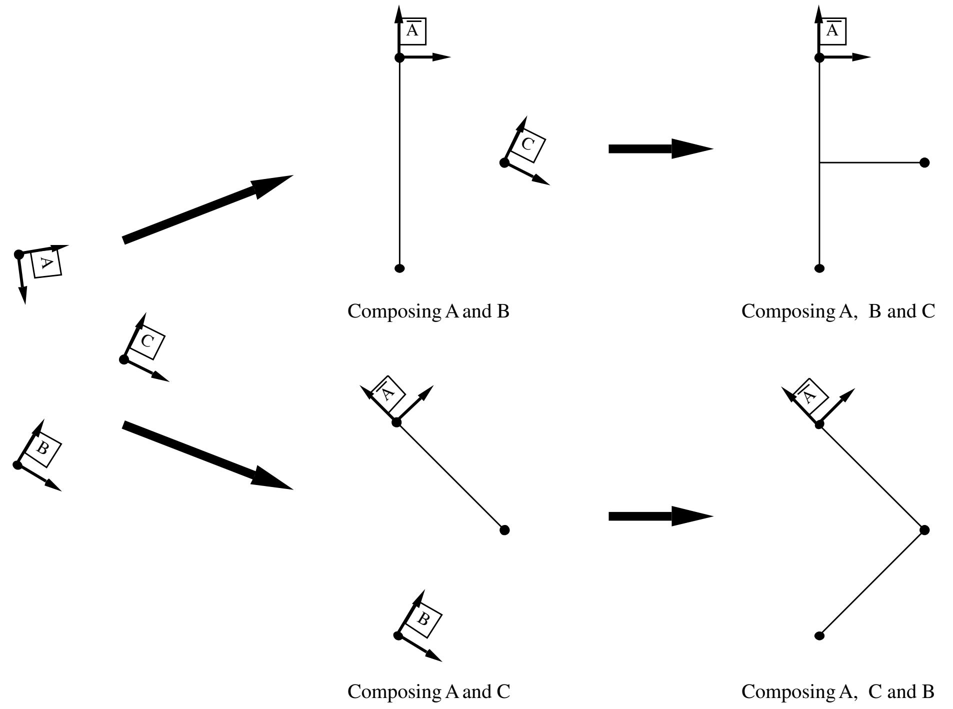


Figure 6.6: The resulting reference depends on the order in which the features are considered

figure 6.6 we can see the result of applying this procedure to a set of three vertices in different order. Thus, in calculating the associated reference of a set of features, the same order must be applied to the observed features and to the model features paired with them.

Given that we are dealing with uncertain geometric information, there is uncertainty in the estimation of the parameters of the relative transformation between the observations, which is due to the uncertainty in the location of the observations. For this reason, in selecting the observations to be used to locate the object, we prefer to use geometric features with low uncertainty, whose estimated relative location is far from the values that make them have common symmetries. For example, in table 6.1 we can see that the common symmetries for two edges are fundamentally determined by their relative angle $\bar{\phi}$. When the estimated value of this angle approaches 0 or $\pm\pi$, or if its covariance is high, the observations should not be used to locate the object. Otherwise, this would result in a very imprecise object-location hypothesis, which in turn makes the constraint validation mechanisms weaker, and results in a less powerful recognition process.

Once we determine that the set of observations have no common symmetries of continuous motion, we can calculate the location of the object. This process is shown in the next section.

6.3.2 Calculating the Object Location

Consider a pair of observations $\{E_1, E_2\}$, whose uncertain locations are represented by $\mathbf{L}_{WE_1} = (\hat{\mathbf{x}}_{WE_1}, \hat{\mathbf{p}}_{E_1}, C_{E_1})$ and $\mathbf{L}_{WE_2} = (\hat{\mathbf{x}}_{WE_2}, \hat{\mathbf{p}}_{E_2}, C_{E_2})$ with respect to reference W, and consider their paired features $\{M_1, M_2\}$, whose location with respect to the object location is represented by $\mathbf{x}_{OM_1}$ and $\mathbf{x}_{OM_2}$. The procedure to determine the location of the object is based on two ideas:

- Since the feature observations may be partial, due to occlusion, the pairing relationship does *not* guarantee that the location of the references associated to the observations and model features coincide (fig. 6.7.a), with the object correctly located. However, the relative transformations $\mathbf{x}_{E_1M_1}$ and $\mathbf{x}_{E_2M_2}$ must belong to the set of symmetries of each feature. This guarantees is that their corresponding aligning references coincide: $\bar{E}_1 = \bar{M}_1$ and $\bar{E}_2 = \bar{M}_2$ (fig. 6.7.b). This is because the geometric relations between the geometric features remain *invariant* under occlusion.

- If the pair of observations have no common symmetries of continuous motion, the object location can be uniquely determined.

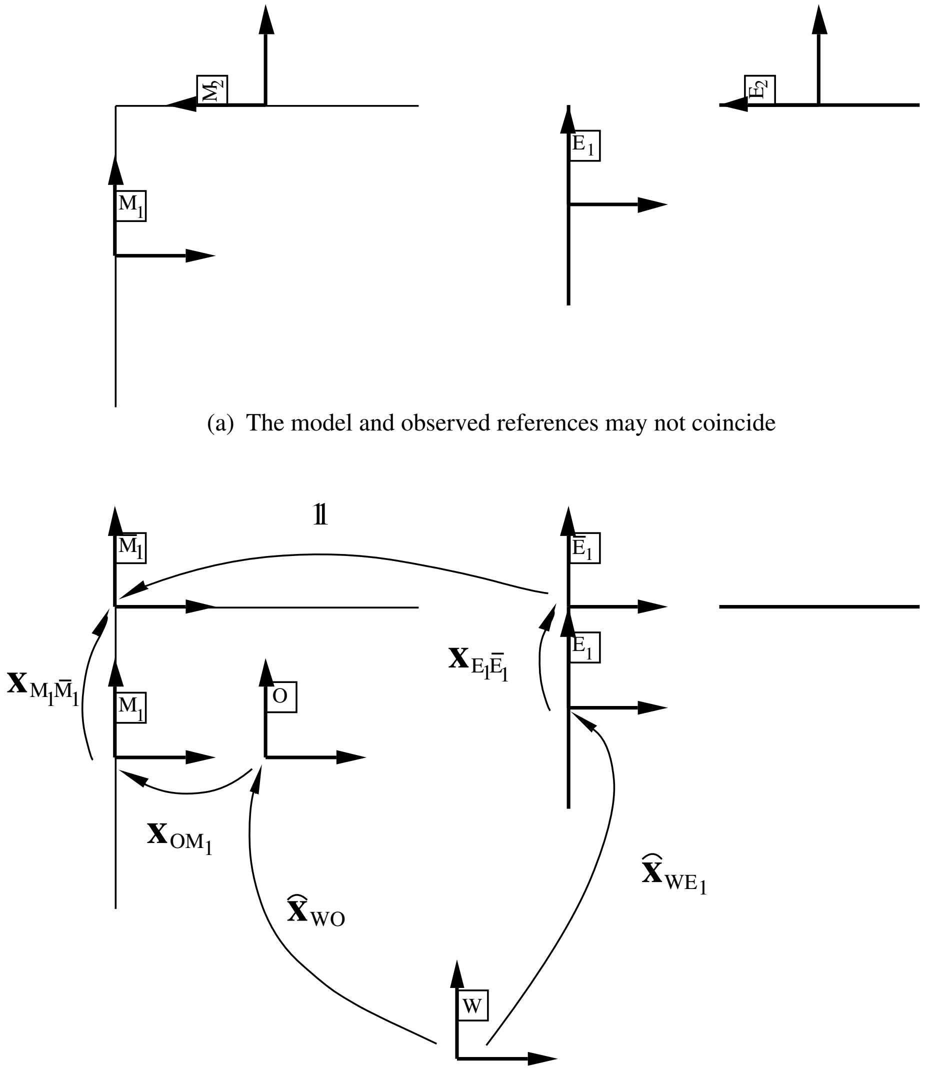
In order to determine the object location we proceed in the following way:

1. Calculate the relative location vectors $\hat{\mathbf{x}}_{E_1E_2}$ and $\mathbf{x}_{M_1M_2}$.

2. Calculate the aligning transformations $\mathbf{x}_{E_1\bar{E}_1}$, $\mathbf{x}_{E_2\bar{E}_2}$, $\mathbf{x}_{M_1\bar{M}_1}$ and $\mathbf{x}_{M_2\bar{M}_2}$.

3. Choose $\bar{E}_1$ and $\bar{M}_1$ to represent the location of the set of observations and features respectively. Since the pairings between the observations and model features implies $E_1 = M_1$, the location of the model in the scene can be calculated as (fig. 6.7.b):

$$\begin{aligned} \hat{\mathbf{x}}_{WO} &= \hat{\mathbf{x}}_{WE_1} \oplus \mathbf{x}_{E_1\bar{E}_1} \ominus \mathbf{x}_{M_1\bar{M}_1} \ominus \mathbf{x}_{OM_1} \\ &= \hat{\mathbf{x}}_{W\bar{E}_1} \ominus \mathbf{x}_{O\bar{M}_1} \end{aligned}$$


M_2
E_2
E_1
M_1
𝟙
$\bar{M}_1$
$\bar{E}_1$
E_1
$\mathbf{X}_{M_1\bar{M}_1}$
$\mathbf{X}_{E_1\bar{E}_1}$
M_1
O
$\mathbf{X}_{OM_1}$
$\widehat{\mathbf{X}}_{WO}$
$\widehat{\mathbf{X}}_{WE_1}$
W

(a) The model and observed references may not coincide

(b) The aligning references coindice and the object location can be determined.

Figure 6.7: The pairing relationship guarantees that the aligning transformations coincide

```
FUNCTION calculate_object_location ($p_1$, $p_2$)

; $p_1 = (e_1, m_1)$, $p_2 = (e_2, m_2)$: pairings
; $p_1 = (\mathbf{L}_{WE_1}, \mathcal{S}_{E_1})$, $e_2 = (\mathbf{L}_{WE_2}, \mathcal{S}_{E_2})$: independent observations
; $m_1 = (\mathbf{x}_{OM_1}, \mathcal{S}_{M_1})$, $m_2 = (\mathbf{x}_{OM_2}, \mathcal{S}_{M_2})$: independent model features
; paired with the observations

; returns an initial estimation of the location of the object $\hat{\mathbf{x}}_{WO}$

    $\hat{\mathbf{x}}_{E_1E_2} := \ominus\hat{\mathbf{x}}_{WE_1} \oplus \hat{\mathbf{x}}_{WE_2}$;
    $\mathbf{x}_{E_1\bar{E}_1}, \mathbf{x}_{E_2\bar{E}_2} :=$ calculate_from_table $\left(\hat{\mathbf{x}}_{E_1E_2}, \mathcal{S}_{E_1}, \mathcal{S}_{E_2}\right)$;
    $\mathbf{x}_{M_1M_2} := \ominus\mathbf{x}_{OM_1} \oplus \mathbf{x}_{OM_2}$;
    $\mathbf{x}_{M_1\bar{M}_1}, \mathbf{x}_{M_2\bar{M}_2} :=$ calculate_from_table $\left(\mathbf{x}_{M_1M_2}, \mathcal{S}_{M_1}, \mathcal{S}_{M_2}\right)$;
    $\hat{\mathbf{x}}_{WO} := \hat{\mathbf{x}}_{WE_1} \oplus \mathbf{x}_{E_1\bar{E}_1} \ominus \mathbf{x}_{M_1\bar{M}_1} \ominus \mathbf{x}_{OM_1}$;

RETURN $\hat{\mathbf{x}}_{WO}$;
END;
```

Algorithm 6.1: *Calculating the Object Location*

> Given that the set of common symmetries of the features is equal to $\mathbb{1}$, the location of the object in the scene is uniquely determined.

Function `calculate_object_location` can be implemented using this procedure (algorithm 6.1). This procedure allows to obtain an *analytic solution* for the location of the object. This initial object location can be used as seed, to linearize the measurement equations, for the object location estimation process, presented in chapter 2. The two observations used to calculate this solution must be included in the estimation afterwards.

Once this problem of calculating the object location from a set of observations is solved, we can concentrate in the problem of selecting the set of observations to be used in the generation of object location hypotheses. This is the subject of the next section.

6.4 Hypothesis Generation

The hypothesis generation process fundamentally consists in selecting a pair of observations, searching in the object models for possible interpretations, and estimating the object location. The selection of an initial pair of observations to generate object-location hypotheses it based in three premises:

- The pair must allow to completely determine the location of the object.
- The number of potential pairings between the pair of observations and model features must be as small as possible.

- The resulting estimation of the object location must be as precise as possible.

The choice of the observations has a considerable impact on the precision of the resulting object location hypothesis. Consider the two dimensional object shown in figure 6.8.a. Suppose we wish to estimate the location of this object in the plane. For this, we obtain information about the location of two vertices of the object (fig. 6.8.b). The following factors have a fundamental impact in the precision of the resulting object location hypothesis:

- *Sensor Precision:* If the precision of the sensor used to obtain the points is greater, evidently this will have a considerable impact in the precision of the object location. (fig. 6.8.c).

- *Feature Location:* Features which are more distant to the object reference have more impact in orientation. Their location with respect to the other observations also determine how correlated position and orientation errors will be. In our example, depending on which vertices we observe, we can obtain a considerably different precision in the estimation of the object location (fig. 6.8.d).

- *Feature Type:* The type of feature we are observing also influences the precision of the object location. If we have observations of more information-contributing features such as edges, we can also obtain higher precision. (fig. 6.8.e).

In systems based on the *identifying before locating* approach, the selection of the observations is solely based on their relevance. This can lead to very imprecise object location hypotheses. Consider the set of observations in figure 6.9. Suppose they come from an instance of the object shown in the figure.

Considering *only the relevance* in the selection of the observations, we would choose them in the following order (longer observed edges will be considered first): $\{E_1, E_3, E_4, E_5, E_2\}$. Supposing we match them correctly, the evolution of the precision of the hypothesis is the one shown in figure 6.10.a. Note that the object location cannot be estimated until we include observation E_5. On the other hand, if we consider the relevance of the observations *and* the precision of the resulting hypothesis, we would choose them in the following order: $\{E_1, E_5, E_3, E_2, E_4\}$. Even if E_3 is more relevant than E_5, we choose E_5 as the second one because it allows to estimate the location of the object. The precision of the hypothesis increases considerably

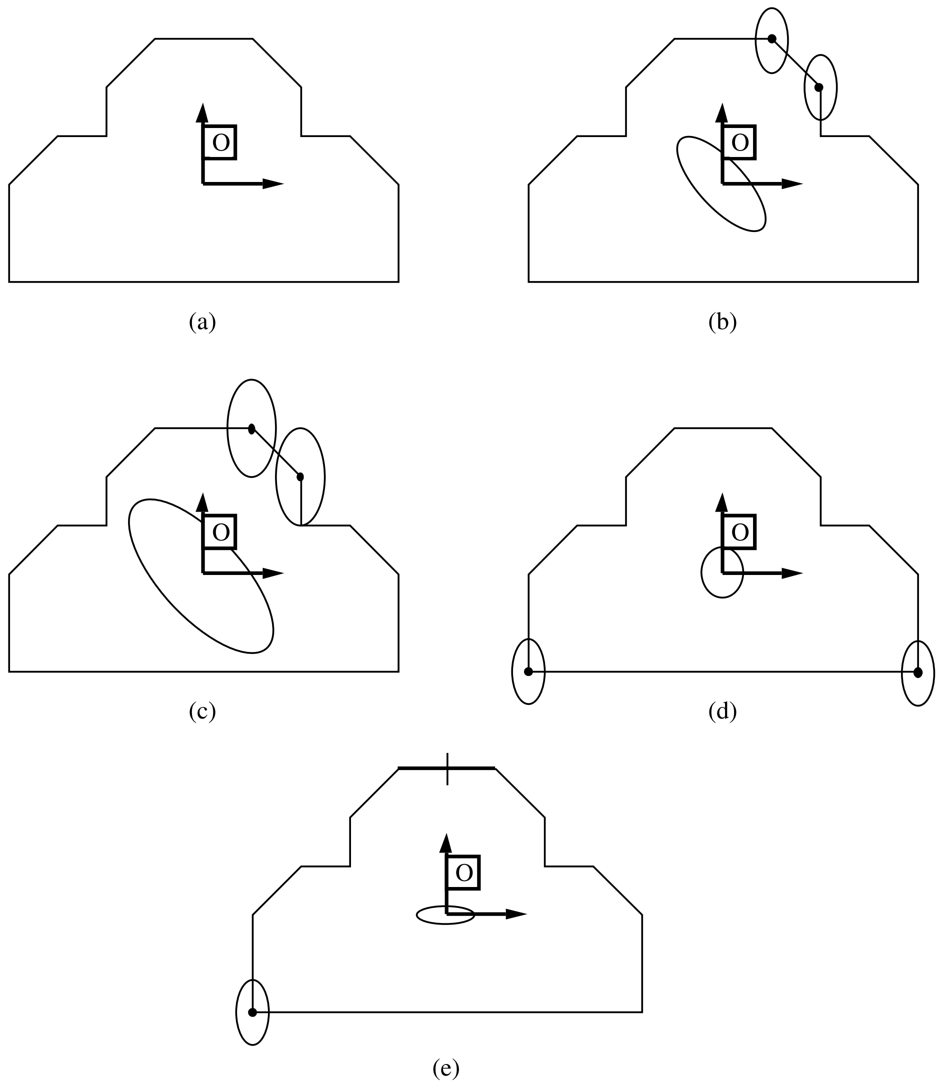


Figure 6.8: Observation selection affects object precision

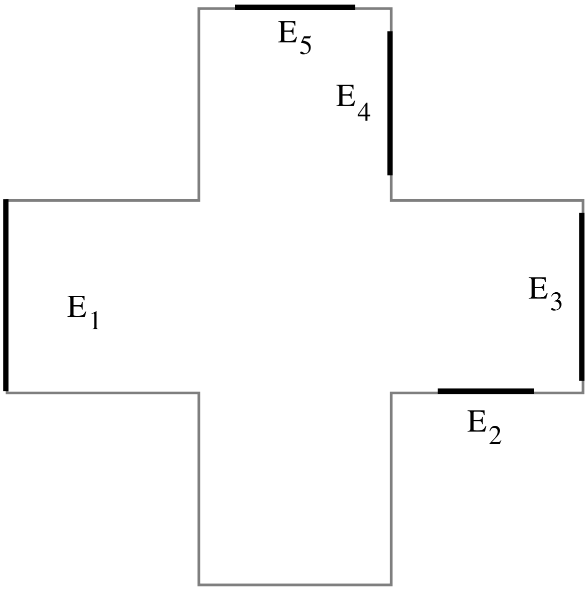


Figure 6.9: Selecting observations to validate a hypothesis

faster. Note that if we take into account both precision and relevance, then the relevance of the hypothesis grows slightly less (figure 6.10.b). This effect is not normally very pronounced, so we can assume it to guarantee a more reliable hypothesis verification process.

In the following sections we propose strategies for the selection of observations that can satisfy the requirements of the hypothesis generation process.

6.4.1 Selecting the First Observation

Given a set $\{e_1, \cdots, e_n\}$ of sensorial observations of geometric features in the scene, the hypothesis generation process selects as first observation the most relevant one. Given that relevance is related to feature size, this observation may have less potential pairings in the models. Additionally, since relevance is also related to the number of d.o.f that determine the feature location, and to the precision of the sensor that obtained the observation, it may allow to generate more precise object location hypotheses, and its may result in more discriminant constraint analysis. Function `select_first_observation` im-

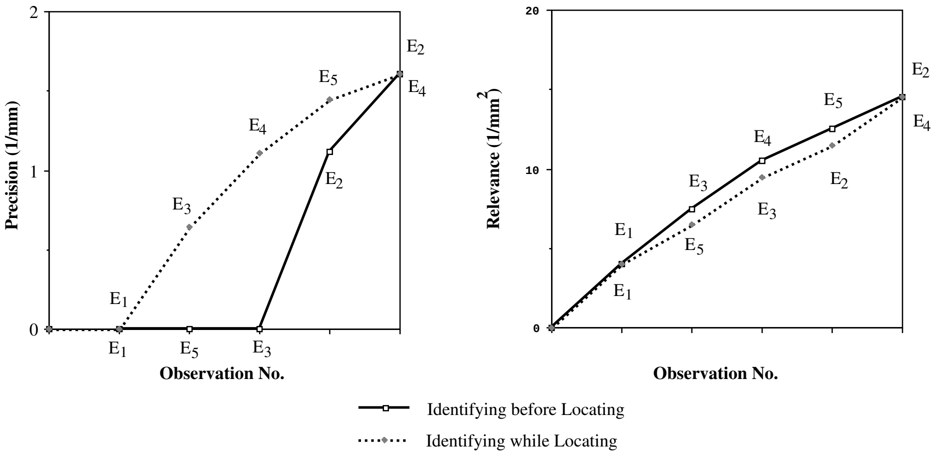


Figure 6.10: Evolution of precision and relevance selecting observations by relevance or by relevance and precision

plements this strategy (see algorithm 6.2).

6.4.2 Selection of the Second Observation

The selection of the second observation is centered around the possibility of obtaining a precise estimation of the object location. For this purpose, we carry out the following procedure (see algorithm 6.2):

1. Select observations whose distance to e_f is smaller than the maximal radius of the objects. This reduces the size of the search space.

2. Order the observations according to their relevance. We will consider each observation to select it as the second observation, and choosing more relevant observations may reduce the number of its potential pairings.

3. Select the first observation, call it e_s and estimate its binary geometric relations with e_f. This allows us to perform symmetries analysis between them to determine whether they have no common symmetries. For this, we consult the table to verify whether the estimated values of their binary relations are far from the values where the two features have common symmetries.

```
FUNCTION select_first_observation ($\mathcal{E}$)

    ; $\mathcal{E}$: set of observations
    ; returns the most relevant observation

    $e := e \mid$ relevance $(e) = \max_{e_i \in \mathcal{E}}$ relevance $(e_i)$;

RETURN $e$;
END;


FUNCTION select_second_observation $\left(\mathcal{E},\, e_f\right)$

    ; $\mathcal{E}$: set of observations
    ; $e_f$: observation selected as first

    $\mathcal{E}_r :=$ choose_proximal_observations ($\mathcal{E}$);
    $\mathcal{E}_r :=$ order_by_relevance ($\mathcal{E}_r$);
    found := FALSE;
    FOR $e_s$ in $\mathcal{E}_r$ WHILE NOT found DO
        $\hat{r},\, C_{\hat{r}} :=$ estimate_geometric_relations ($e_f$, $e_s$);
        IF NOT common_symmetries ($\mathcal{S}_A$, $\mathcal{S}_B$, $\hat{r}$) THEN
            $\hat{\mathbf{x}}_{WO} := \hat{\mathbf{x}}_{W\hat{E}_f}$;
            $\hat{\mathbf{d}}_O := 0$;
            $C_O := \left(B^T_{\bar{E}_f} C^{-1}_{\bar{E}_f} B_{\bar{E}_f} + J^T_{\bar{E}_f \bar{E}_s} B^T_{\bar{E}_s} C^{-1}_{\bar{E}_s} B_{\bar{E}_s} J_{\bar{E}_f \bar{E}_s}\right)^{-1}$;
            found := precision ($\mathbf{L}_{WO}$) $\geq$ threshold ;
        FI;
    OD;

RETURN $e_s$;
END;
```

Algorithm 6.2: *Selecting Observations*

4. We are also interested in obtaining a precise estimation of the object location. In order to estimate the precision of the resulting object location hypothesis, we can calculate the uncertainty in the location of reference $\bar{E}_f$ due to the uncertainty in the location of the two observation e_f and e_s, which approximately corresponds to the uncertainty of the object location if its associated reference O coincides with $\bar{E}_f$:

$$C_O \approx C_{\bar{E}_f} = \left(B^T_{\bar{E}_f} C^{-1}_{\bar{E}_f} B_{\bar{E}_f} + J^T_{\bar{E}_f \bar{E}_s} B^T_{\bar{E}_s} C^{-1}_{\bar{E}_s} B_{\bar{E}_f} J_{\bar{E}_s \bar{E}_f}\right)^{-1}$$

From this covariance matrix we can determine how precise the resulting object location hypothesis will be.

5. If the selected observation e_s does not satisfy this requirement, we discard it and go back to step 3.

We can use $\hat{r}$, the estimated geometric relations between e_f and e_s, along with the characteristics of e_f and e_s to search for correspondences in the models. This is done by searching for all model features of the types of e_f and e_s whose geometric relations correspond to $\hat{r}$. The geometric relations between model features can be precomputed in tables that make this search more efficient. From these pairs of model features, we select the ones which satisfy unary constraints with e_f and e_s.

This yields a set of competing object-location hypotheses $h = \{h_1, \cdots, h_l\}$ which are supported by e_f and e_s. We can choose to generate only a small set of hypotheses for verification, and if the solution is not found among them, more hypotheses can be generated.

6.5 Hypothesis Verification

There is a potentially large set of hypotheses that can explain the selected pair of observations. In the hypothesis verification process we are faced with the problem of selecting the order in which we consider the hypotheses, and the way in which these should be verified. Our hypothesis verification process is composed of three steps:

- We choose to verify the most precise hypotheses first, because their verification may be more reliable and less costly.

- If there is more available sensor information, the verification of a hypothesis is carried out in a *data-driven* manner, in which the system tries to incorporate the available information.

- Once there is no more available information, further verification and refinement of a hypothesis is carried out in a *model-driven* manner, in which the system predicts the location of a feature in the scene, and selects a sensor and sensor location to verify its presence.

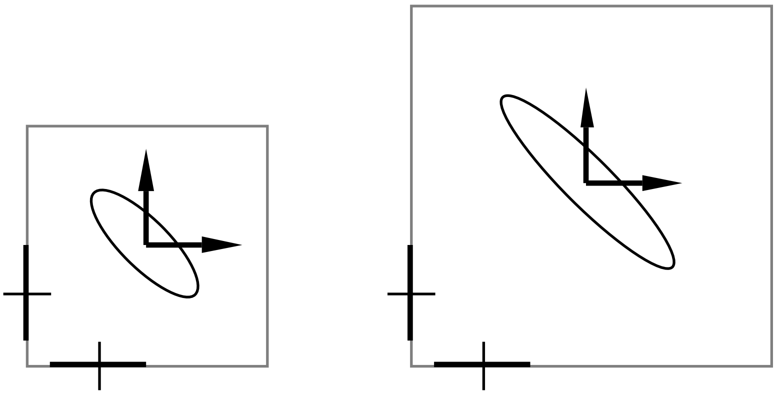


Figure 6.11: Alternative hypotheses may have different precision

6.5.1 Hypothesis Selection

Alternative hypotheses for the same set of observations may have different precision (fig. 6.11), and the precision of the estimation of the object location has a great impact in the cost and reliability of its verification. As we have seen in the previous chapter, the verification of the pairings between the predicted features and observations is carried out using a statistical test that validates whether their corresponding locations coincide. The precision of the predicted location of the model feature affects the discriminancy of this test because the larger the volume of uncertainty of the feature location, the larger the volume of acceptance of the test (fig. 6.12). This means that model features with imprecise predicted location have a higher probability of being spuriously paired. Additionally, the volume of uncertainty is related to the cost of finding the feature because it defines the size of the region in the scene in which the feature must be searched for.

6.5.2 Data-Driven Verification

If there are available observations within the region hypothetically occupied by the object, the system may try to pair them with features in the model. In this early verification step, augmenting the credibility of the hypothesis is the central goal, so observations are taken in order of relevance.

We use the fundamental data-driven verification algorithm 1.5 of chapter 1 using the `select_first_observation` function of algorithm 6.2 to select

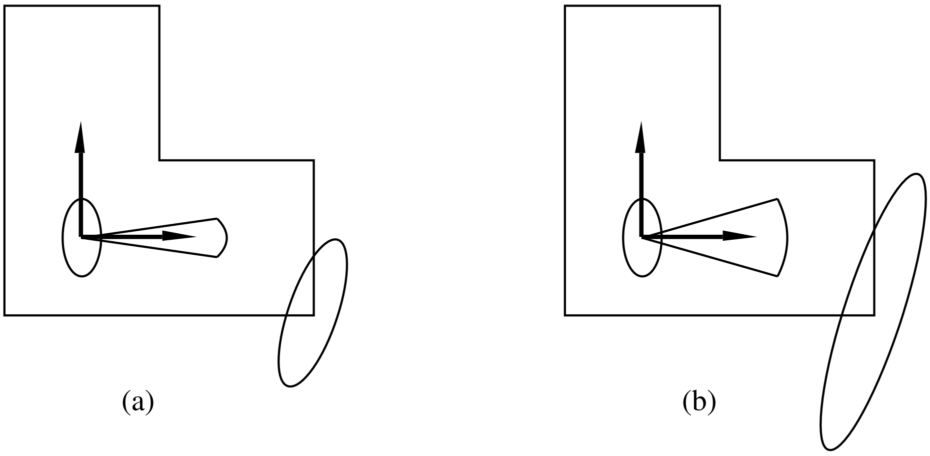


Figure 6.12: Precise hypotheses (a) are less costly to verify than (b) imprecise hypotheses

the most relevant observation. Given the estimated location of the observation e_i, and the estimated location of the object according to hypothesis h_j, we estimate the location of the observation relative to the object location, and search in the model for a feature whose relative location to the object may be compatible. If such a model feature exists, and the feature and the observations satisfy the rigidity and extension constraints, the pairing is established and the observation is integrated into the estimated location of the object.

The available set of observations can be limited to the ones that are within the region hypothetically occupied by the object.

6.5.3 Model-Driven Verification

Once there is no more available sensor information, a hypothesis may be further verified using a prediction-verification scheme, in which the location of features in the scene is predicted and some sensor is selected to try to observe it (see algorithm 6.3). The central issue of this prediction-verification scheme is the selection of the model feature to be predicted. Our approach consists in selecting the feature that potentially contributes most to the precision of the hypothesis, until a sufficient level of hypothesis precision is attained. Then, the potential relevance will be used in the selection of the

```
FUNCTION verify_hypothesis ($h$, $\mathcal{M}$)

; $h = (\mathbf{L}_{WO}, \mathcal{S}_h)$: object-location hypothesis to verify
; $\mathcal{M}$: set of model features

; Selects a model feature and predicts its location in the scene
; selects a sensor and its location to verify the feature


   WHILE relevance($h$) $<$ $threshold_R$ $<$ relevance($h$)$+$ total($\mathcal{M}$) DO
      $m :=$ select_model_feature ($h$, $\mathcal{M}$);
      $\mathbf{L}_{WM} := \mathbf{L}_{WO} \oplus \mathbf{x}_{OM}$;
      $e :=$ verify_feature ($\mathbf{L}_{WM}$);
      IF $e \neq$ null THEN
         $p := (e, m)$;
         $\mathcal{S}_h := \mathcal{S}_h \cup \{p\}$;
         $\mathbf{L}_{WO} :=$ refine_object_location ($\mathbf{L}_{WO}$, $p$);
      FI;
      $\mathcal{M} := \mathcal{M} \setminus \{m\}$;
   OD;

RETURN $h$;
END;


FUNCTION select_model_feature($h$, $\mathcal{M}$)

; $h = (\mathbf{L}_{WO}, \mathcal{S}_h)$: object-location hypothesis to verify
; $\mathcal{M}$: set of model features

; decides if precision or relevance is the goal of verification and then
; selects a model feature to maximize the selected goal

   IF precision ($\mathbf{L}_{WO}$) $>$ $threshold_P$ THEN
      $m := m \mid$ potential_relevance ($m$) $= \max_{m_i \in \mathcal{M}}$ potential_relevance ($m_i$);
   ELSE
      $c :=$ most_uncertain_component ($\mathbf{L}_{WO}$);
      $m := m \mid$ potential_contribution ($m$, $c$) $= \max_{m_i \in \mathcal{M}}$ potential_contribution ($m_i$, $c$);
   FI;

RETURN $m$;
END;
```

Algorithm 6.3: *Model-driven Hypothesis Verification*

feature to predict, so that the relevance of the hypothesis will be augmented. In the selection of the sensor and the computation of its location, we take into account which component of the object location is less precise, in order to direct the sensor to augment this precision, as explained in chapter 5.

Function `select_model_feature` implements this model feature selection strategy (see algorithm 6.3). Function `potential_relevance` calculates the potential relevance of the model feature, as explained in chapter 4. The purpose of function `potential_contribution` is to calculate the potential contribution of the model feature to the object location, as explained in chapter 5.

6.6 Conclusions

In this chapter we have shown the advantages of having an early estimation of object location in speeding up the recognition process. Essentially it allows to validate a smaller set of tighter geometric constraints. We have given a general procedure to determine whether a set of observations allow to completely determine the object location and to calculate this location.

We have also seen how the selection of the observations to be used in the hypotheses generation process has great influence in the precision of the estimated object location, and thus, in the cost of the hypothesis verification process. We have also seen that even though identification and localization are not equivalent processes, they can mutually benefit in a simultaneous scheme. The observation selection strategies given here are designed to obtain more precise object-location hypothesis, and thus, limit the probability of accepting an incorrect interpretation of the observations.

The hypothesis verification process proposed here is a two step process in which the system first tries to make use of the available set of observations in a *data-driven* manner. Once there is no more available sensorial information, hypothesis verification proceeds in a *model-driven* manner, predicting the location of features in the scene and selecting a sensor and sensor location to verify their presence. The system takes into account the precision and relevance of the object hypothesis to select the feature that may help attain rapid recognition.

Conclusions

In this work we have carried out a comparative study of the two fundamental schemes of geometric object recognition: the *identifying before locating* scheme, in which the identity of an object is determined, and then its location in the scene estimated, and the *identifying while locating* scheme, in which identification and localization are simultaneous processes. We contribute to solve some fundamental problems that are common to both recognition schemes, and also some that are specific of each one. Our main contributions are:

- The definition of general and powerful *validation methods* that allow to quickly discard inconsistent interpretations.
- The specification of *perception strategies* directed towards the obtention of additional sensorial information with the purpose of obtaining more precise hypotheses.
- The definition of *recognition strategies* that allow the system to search more efficiently for plausible interpretations.

Our efforts have been concentrated on the development of methods that are *not* based on a specific type of geometric feature or sensor, and for this reason these methods are suitable for multisensor systems.

We use the SPmodel, a probabilistic model for the representation of uncertain geometric information. This model can be used to represent the location of any type of geometric entity and its uncertainty. From a theoretic and from a practical point of view, it seems a more adequate representation for uncertain locations than set-based models. This model has been used to establish a general integration mechanism for uncertain geometric information, based on the Extended Kalman and information filters. In chapter 2, we use a nonrecursive version of the Extended information filter to obtain observation models for different types of geometric elements given by two types of sensors: mobile proximity and mobile 2D vision. These observation models are used as illustrative examples of the generality of our approach.

In chapter 3, we propose a general method to compute and validate geometric constraints between uncertain geometric features. This procedure allows to derive both location independent and location dependent geometric relations for all types of geometric features in a systematic way. Geometric constraints are validated using a uniform procedure which explicitly considers the uncertainty associated to the use of different sensors. This constraint validation mechanism allows to discard regions of the solution space which only contain inconsistent interpretations of the available sensorial information.

In chapter 4, we define two measurements used to characterize the relevance and precision of an object-location hypothesis, as a function of the set of observation-model pairings that support the hypothesis. The precision of the estimated location of the object allows to determine which feature may make this estimation more precise. The relevance of an observation is a general measurement that allows to compare the contribution of different types of geometric features to the identification of the object. These measurements allow us to define a set of strategies for the selection of observations and model features that allow the system to search efficiently in the solution space. Such strategies are described in chapter 6.

In chapter 5 we propose a set of perception strategies with the twofold goal of *verifying* an object-location hypothesis and *refining* its estimated location. Obtaining a precise object location allows constraint validation mechanisms to be more discriminant. Such perception strategies determine which feature of an object can contribute more location information, select the most suitable sensor for a given perception task, and compute its location so that its contribution of location information is maximal.

In chapter 6, we show the advantages of the *identifying while locating* recognition scheme. From the constraint validation point of view, the availability of the estimated object location allows to apply location dependent constraints, which are more discriminant and less costly validation mechanisms than location independent constraints. Additionally, the probability of generating an incorrect interpretation of the sensorial data is reduced, and the availability of *a priori* information for sensors allows the use of less costly and more reliable perception processes.

This work is fundamentally a theoretical analysis of the recognition problem. Thus, our main concerns in the present and near future are the implementation of the perception and recognition strategies described here. For this purpose, several undergraduate projects are being carried out, and we will soon be able to report their results.

Some aspects of the recognition problem have been considered from a restricted point of view. One is the use of perception strategies for the verification of an object location hypothesis. A generalization is required, in which the system may consider several hypotheses simultaneously, and can determine which feature will allow to disambiguate between them. Another issue to be solved is the evaluation of the relevance of a geometric feature to recognition taking into account possible similarities between objects, and its capacity in discriminating among them. There are also several aspects of recognition that remain to be dealt with, such as *termination*: deciding when the object recognition process has obtained a suitable interpretation, and when it will not be possible to obtain one. The termination mechanism must guarantee that an incorrect interpretation will not be given as solution, and also limit the probability of missing a correct one.

In the long term, we wish to undertake more general problems, such as recognition in situations where there is less *a priori* information about the scene. A very interesting problem of this type is the exploration of indoor environments using a mobile robot. This will constitute our area of future research.

Appendix A

Transformations and Jacobian Matrices

A.1 Homogeneous Matrices and Location Vectors

In this work we use two alternative representations for transformations: *homogeneous matrices* and *location vectors* formed by three cartesian coordinates and three Roll-Pitch-Yaw angles. In this section we summarize their fundamental properties and laws of transformation between the two representations. More complete information can be found in [Paul 81].

A homogeneous matrix H is a 4×4 matrix of the form:

$$H = \begin{bmatrix} R & \mathbf{p} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where: R is a 3×3 orthogonal rotation matrix: $R^{-1} = R^T$

$\mathbf{p}$ is a 3-dimensional translation vector

$\mathbf{n}, \mathbf{o}, \mathbf{a}$ are the three column vectors that form R

The representation of the elementary transformations using homogeneous matrices is the following:

$$t = Trasl(x, y, z) \qquad H = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$t = Rot(x, \psi) \qquad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ 0 & \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$t = Rot(y, \theta) \qquad H = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$t = Rot(z, \phi) \qquad H = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformation composition and inversion is equivalent to homogeneous matrix product and inversion. Due to the special form of these matrices, these operations can be carried out as follows:

$$H_3 = H_1 H_2 = \begin{bmatrix} R_1 R_2 & \mathbf{p}_1 + R_1 \mathbf{p}_2 \\ 0 & 1 \end{bmatrix}$$

$$H^{-1} = \begin{bmatrix} R^T & -R^T \mathbf{p} \\ 0 & 1 \end{bmatrix}$$

An RPY location vector is composed of three cartesian coordinates and three Roll-Pitch-Yaw angles. The form of this vector and the transformation it represents are:

$$\begin{aligned} &\mathbf{x} = (x,\ y,\ z,\ \psi,\ \theta,\ \phi)^T \\ &\phi : Roll \quad , \quad \theta : Pitch \quad , \quad \psi : Yaw \\ &t = Trasl(x, y, z)\ Rot(z, \phi)\ Rot(y, \theta)\ Rot(x, \psi) \end{aligned}$$

The conversions between location vectors and homogeneous matrices are given by:

$$H = Hom(\mathbf{x}) =$$

$$\begin{bmatrix} \cos\phi\cos\theta & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & x \\ \sin\phi\cos\theta & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & y \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{x} = Loc(H) = \begin{pmatrix} x \\ y \\ z \\ \psi \\ \theta \\ \phi \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \\ p_z \\ atan2(o_z, a_z) \\ atan2(-n_z, n_x\cos\phi + n_y\sin\phi) \\ atan2(n_y, n_x) \end{pmatrix}$$

The submatrix of $Hom(\mathbf{x})$ corresponding to the rotation matrix will be represented by:

$$R = Mrot(\mathbf{x})$$

We shall represent the composition and inversion of location vectors by means of operators $\oplus$ and $\ominus$ respectively, following Smith's notation [Smith 88]. To calculate them, location vectors shall be converted to homogeneous matrices, these shall be multiplied or inverted, and the opposite conversion shall be carried out:

$$\begin{array}{l} \mathbf{x}_3 = \mathbf{x}_1 \oplus \mathbf{x}_2 = Loc(Hom(\mathbf{x}_1)Hom(\mathbf{x}_2)) \\ \ominus\mathbf{x}_1 = Loc((Hom(\mathbf{x}_1))^{-1}) \end{array}$$

If there are compositions followed by inversions, these shall be abbreviated by:

$$\mathbf{x}_1 \ominus \mathbf{x}_2 = \mathbf{x}_1 \oplus (\ominus\mathbf{x}_2)$$

From homogeneous matrices, we can obtain a direct graphical interpretation for transformations (fig. A.1). The fundamental properties of this representation, which are used throughout this work to derive most of the presented results are:

1. Being R an orthonormal matrix, we have $\|\mathbf{n}\| = \|\mathbf{o}\| = \|\mathbf{a}\| = 1$, and thus:

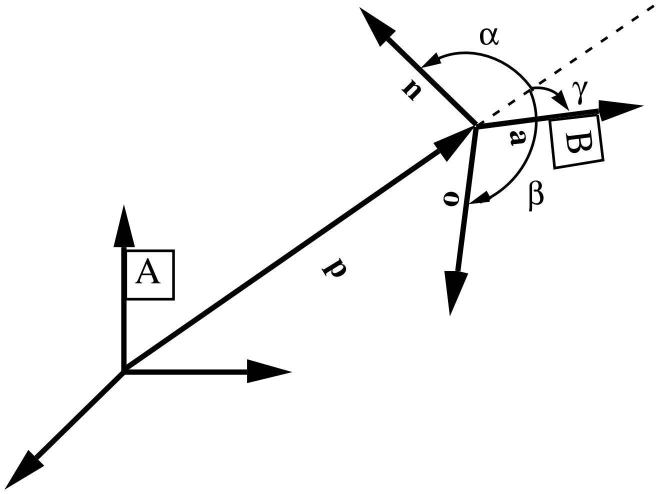


Figure A.1: Graphical representation of the homogeneous matrix H_{AB}

$$
\begin{aligned}
n_x^2 + n_y^2 + n_z^2 &= 1\\
o_x^2 + o_y^2 + o_z^2 &= 1\\
a_x^2 + a_y^2 + a_z^2 &= 1\\
n_x^2 + o_x^2 + a_x^2 &= 1\\
n_y^2 + o_y^2 + a_y^2 &= 1\\
n_z^2 + o_z^2 + a_z^2 &= 1
\end{aligned} \tag{A.1}
$$

2. In the same way, $\mathbf{n} \perp \mathbf{o}$, $\mathbf{o} \perp \mathbf{a}$, and $\mathbf{n} \perp \mathbf{a}$. Thus:

$$
\mathbf{n} \times \mathbf{o} = \mathbf{a} \implies \begin{aligned} a_x &= n_y\, o_z - n_z\, o_y\\ a_y &= n_z\, o_x - n_x\, o_z\\ a_z &= n_x\, o_y - n_y\, o_x \end{aligned} \tag{A.2}
$$

$$
\begin{aligned}
(\mathbf{p} \times \mathbf{n}) \cdot \mathbf{o} = \mathbf{p} \cdot \mathbf{a} \quad &; \quad (\mathbf{p} \times \mathbf{o}) \cdot \mathbf{n} = -\mathbf{p} \cdot \mathbf{a}\\
(\mathbf{p} \times \mathbf{o}) \cdot \mathbf{a} = \mathbf{p} \cdot \mathbf{n} \quad &; \quad (\mathbf{p} \times \mathbf{a}) \cdot \mathbf{o} = -\mathbf{p} \cdot \mathbf{n}\\
(\mathbf{p} \times \mathbf{a}) \cdot \mathbf{n} = \mathbf{p} \cdot \mathbf{o} \quad &; \quad (\mathbf{p} \times \mathbf{n}) \cdot \mathbf{a} = -\mathbf{p} \cdot \mathbf{o}
\end{aligned} \tag{A.3}
$$

3. Let $\mathbf{p} \times \mathbf{n} = (pn_x,\, pn_y,\, pn_z)^T$, $\mathbf{p} \times \mathbf{o} = (po_x,\, po_y,\, po_z)^T$, and $\mathbf{p} \times \mathbf{a} = (pa_x,\, pa_y,\, pa_z)^T$. We have:

$$\begin{aligned}
pn_x &= p_y\, n_z - p_z\, n_y \\
pn_y &= p_z\, n_x - p_x\, n_z \\
pn_z &= p_x\, n_y - p_y\, n_x \\
\\
po_x &= p_y\, o_z - p_z\, o_y \\
po_y &= p_z\, o_x - p_x\, o_z \\
po_z &= p_x\, o_y - p_y\, o_x \\
\\
pa_x &= p_y\, a_z - p_z\, a_y \\
pa_y &= p_z\, a_x - p_x\, a_z \\
pa_z &= p_x\, a_y - p_y\, a_x
\end{aligned} \tag{A.4}$$

We have:

$$\begin{aligned}
pn_x^2 + po_x^2 + pa_x^2 &= (p_y\, n_z - p_z\, n_y)^2 + (p_y\, o_z - p_z\, o_y)^2 + (p_y\, a_z - p_z\, a_y)^2 \\
&= \left(n_z^2 + o_z^2 + a_z^2\right) p_y^2 + \left(n_y^2 + o_y^2 + a_y^2\right) p_z^2 \\
&\quad -2\,(n_y\, n_z + o_y\, o_z + a_y\, a_z)\, p_y\, p_z \\
&= p_y^2 + p_z^2
\end{aligned}$$

Similarly:

$$\begin{aligned}
pn_y^2 + po_y^2 + pa_y^2 &= p_x^2 + p_z^2 \\
pn_z^2 + po_z^2 + pa_z^2 &= p_x^2 + p_y^2
\end{aligned} \tag{A.5}$$

Thus, we have:

$$\begin{aligned}
\|\mathbf{p} \times \mathbf{n}\|^2 + \|\mathbf{p} \times \mathbf{o}\|^2 + \|\mathbf{p} \times \mathbf{a}\|^2 &= pn_x^2 + po_x^2 + pa_x^2 \\
&\quad + pn_y^2 + po_y^2 + pa_y^2 \\
&\quad + pn_z^2 + po_z^2 + pa_z^2 \\
&= p_y^2 + p_z^2 + p_x^2 + p_z^2 + p_x^2 + p_y^2 \\
&= 2\,\|\mathbf{p}\|^2
\end{aligned} \tag{A.6}$$

4. From Lagrange's Identity, which states that given two vectors $\mathbf{v}$ and $\mathbf{w}$:

$$(\mathbf{v} \cdot \mathbf{w})^2 = \|\mathbf{v}\|^2 \|\mathbf{w}\|^2 - \|\mathbf{v} \times \mathbf{w}\|^2$$

we have:

$$\begin{aligned}(\mathbf{p} \cdot \mathbf{n})^2 + (\mathbf{p} \cdot \mathbf{o})^2 + (\mathbf{p} \cdot \mathbf{a})^2 &= \|\mathbf{p}\|^2 \|\mathbf{n}\|^2 + \|\mathbf{p}\|^2 \|\mathbf{o}\|^2 + \|\mathbf{p}\|^2 \|\mathbf{a}\|^2 \\ &\quad - \|\mathbf{p} \times \mathbf{n}\|^2 - \|\mathbf{p} \times \mathbf{o}\|^2 - \|\mathbf{p} \times \mathbf{a}\|^2 \\ &= 3\,\|\mathbf{p}\|^2 - 2\,\|\mathbf{p}\|^2 \\ &= \|\mathbf{p}\|^2 \qquad \text{(A.7)}\end{aligned}$$

5. Given that (see fig. A.1):

$$\begin{aligned}\|\mathbf{p} \times \mathbf{n}\|^2 &= \|\mathbf{p}\|^2 \sin^2 \alpha \\ \|\mathbf{p} \times \mathbf{o}\|^2 &= \|\mathbf{p}\|^2 \sin^2 \beta \\ \|\mathbf{p} \times \mathbf{a}\|^2 &= \|\mathbf{p}\|^2 \sin^2 \gamma\end{aligned}$$

we have:

$$\sin^2 \alpha + \sin^2 \beta + \sin^2 \gamma = 2 \qquad \text{(A.8)}$$

6. Given that (see fig. A.1):

$$\begin{aligned}\mathbf{p} \cdot \mathbf{n} &= \|\mathbf{p}\| \cos \alpha \\ \mathbf{p} \cdot \mathbf{o} &= \|\mathbf{p}\| \cos \beta \\ \mathbf{p} \cdot \mathbf{a} &= \|\mathbf{p}\| \cos \gamma\end{aligned}$$

we have:

$$\cos^2 \alpha + \cos^2 \beta + \cos^2 \gamma = 1 \qquad \text{(A.9)}$$

A.2 Jacobians of the Composition

The Jacobians of the composition of location vectors, with respect to the first and second operand are given by: [Smith 88]:

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

Their value can be calculated using the following formulas, extracted from [Smith 88] [1]

$$J_{1\oplus}\{\mathbf{x}_1, \mathbf{x}_2\} = \begin{bmatrix} I_{3\times 3} & M \\ 0_{3\times 3} & K \end{bmatrix} \qquad J_{2\oplus}\{\mathbf{x}_1, \mathbf{x}_2\} = \begin{bmatrix} R_1 & 0_{3\times 3} \\ 0_{3\times 3} & K' \end{bmatrix}$$

where, taking $\mathbf{x}_3 = \mathbf{x}_1 \oplus \mathbf{x}_2$:

$$M = \begin{bmatrix} y_2 a_{x_1} - z_2 o_{x_1} & (z_3 - z_1)\cos\phi_1 & y_1 - y_3 \\ y_2 a_{y_1} - z_2 o_{y_1} & (z_3 - z_1)\sin\phi_1 & x_3 - x_1 \\ y_2 a_{z_1} - z_2 o_{z_1} & -x_2\cos\theta_1 - y_2\sin\theta_1\sin\psi_1 - z_2\sin\theta_1\cos\psi_1 & 0 \end{bmatrix}$$

$$K = \begin{bmatrix} \cos\theta_1\cos(\phi_3 - \phi_1)/\cos\theta_3 & \sin(\phi_3 - \phi_1)/\cos\theta_3 & 0 \\ -\cos\theta_1\sin(\phi_3 - \phi_1) & \cos(\phi_3 - \phi_1) & 0 \\ (o_{x_2}\sin\psi_3 + a_{x_2}\cos\psi_3)/\cos\theta_3 & \sin\theta_3\sin(\phi_3 - \phi_1)/\cos\theta_3 & 1 \end{bmatrix}$$

$$K' = \begin{bmatrix} 1 & \sin\theta_3\sin(\psi_3 - \psi_2)/\cos\theta_3 & (a_{x_1}\cos\phi_3 + a_{y_1}\sin\phi_3)/\cos\theta_3 \\ 0 & \cos(\psi_3 - \psi_2) & -\cos\theta_2\sin(\psi_3 - \psi_2) \\ 0 & \sin(\psi_3 - \psi_2)/\cos\theta_3 & \cos\theta_2\cos(\psi_3 - \psi_2)/\cos\theta_3 \end{bmatrix}$$

$R_1 = Mrot(\mathbf{x}_1)$

It can be seen that the expression of the formulas given above has been simplified using terms of $\mathbf{x}_3 = \mathbf{x}_1 \oplus \mathbf{x}_2$ and terms of the rotation matrix corresponding to $\mathbf{x}_1$ y $\mathbf{x}_2$. In the case where the value of θ_3 be $-\pi/2$ or $\pi/2$, the Jacobians would be undefined. These values correspond to singular configurations of Roll-Pitch-Yaw angles.

[1] Smith's formulas have rows 4 and 6 permuted, due to the fact that the location vectors used in such work are of the form $\mathbf{x} = (x,\ y,\ z,\ \phi,\ \theta,\ \psi)^T$, with terms 4 and 6 permuted with respect to our representation.

In the case where the term with respect to which it is being derived equals zero (identity transformation), the expressions of the Jacobians are greatly simplified:

$$J_{1\oplus}\{0,\mathbf{x}\} = \begin{bmatrix} I_{3\times3} & M \\ 0_{3\times3} & K \end{bmatrix} \qquad J_{2\oplus}\{\mathbf{x},0\} = \begin{bmatrix} R & 0_{3\times3} \\ 0_{3\times3} & K' \end{bmatrix}$$

where:

$$M = \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix}$$

$$K = \begin{bmatrix} \cos\phi/\cos\theta & \sin\phi/\cos\theta & 0 \\ -\sin\phi & \cos\phi & 0 \\ \sin\theta\cos\phi/\cos\theta & \sin\theta\sin\phi/\cos\theta & 1 \end{bmatrix}$$

$$K' = \begin{bmatrix} 1 & \sin\theta\sin\psi/\cos\theta & \sin\theta\cos\psi/\cos\theta \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi/\cos\theta & \cos\psi/\cos\theta \end{bmatrix}$$

$$R = Mrot(\mathbf{x})$$

The calculation of the inverse of the Jacobians of the composition can also be carried out in a simple way, without the need of inverting matrices, making use of the following equalities:

$$\begin{aligned} J_{1\oplus}^{-1}\{\mathbf{x}_1,\mathbf{x}_2\} &= \left[\left. \frac{\partial(\mathbf{y}\oplus\mathbf{z})}{\partial\mathbf{y}} \right|_{\mathbf{y}=\mathbf{x}_1,\ \mathbf{z}=\mathbf{x}_2} \right]^{-1} \\ &= \left. \frac{\partial\mathbf{y}}{\partial(\mathbf{y}\oplus\mathbf{z})} \right|_{\mathbf{y}=\mathbf{x}_1,\ \mathbf{z}=\mathbf{x}_2} \\ &= \left. \frac{\partial[(\mathbf{y}\oplus\mathbf{z})\oplus(\ominus\mathbf{z})]}{\partial(\mathbf{y}\oplus\mathbf{z})} \right|_{\mathbf{y}\oplus\mathbf{z}=\mathbf{x}_1\oplus\mathbf{x}_2,\ \ominus\mathbf{z}=\ominus\mathbf{x}_2} \\ &= J_{1\oplus}\{\mathbf{x}_1\oplus\mathbf{x}_2,\ \ominus\mathbf{x}_2\} \\ J_{2\oplus}^{-1}\{\mathbf{x}_1,\mathbf{x}_2\} &= \left. \frac{\partial\mathbf{z}}{\partial(\mathbf{y}\oplus\mathbf{z})} \right|_{\mathbf{y}=\mathbf{x}_1,\ \mathbf{z}=\mathbf{x}_2} \\ &= \left. \frac{\partial[(\ominus\mathbf{y})\oplus(\mathbf{y}\oplus\mathbf{z})]}{\partial(\mathbf{y}\oplus\mathbf{z})} \right|_{\ominus\mathbf{y}=\ominus\mathbf{x}_1,\ \mathbf{y}\oplus\mathbf{z}=\mathbf{x}_1\oplus\mathbf{x}_2} \\ &= J_{2\oplus}\{\ominus\mathbf{x}_1,\ \mathbf{x}_1\oplus\mathbf{x}_2\} \end{aligned}$$

In some expressions, the inverse of $J_{2\oplus}\{\mathbf{x},0\}$ appears. This inverse can be calculated using the previous equality, or in a much simpler way, by:

$$J_{2\oplus}^{-1}\{\mathbf{x},0\} = \begin{pmatrix} R^T & 0_{3\times 3} \\ 0_{3\times 3} & K'^{-1} \end{pmatrix}$$

where:

$$K'^{-1} = \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\psi & \cos\theta\sin\psi \\ 0 & -\sin\psi & \cos\theta\cos\psi \end{pmatrix}$$
$$R = Mrot(\mathbf{x})$$

A.3 Jacobian of the Inversion

The Jacobian of the inverse of a location vector is given by:

$$J_{\ominus}\{\mathbf{x}\} = \left.\frac{\partial(\ominus\mathbf{y})}{\partial\mathbf{y}}\right|_{\mathbf{y}=\mathbf{x}}$$

Its value can be obtained using the following formulas, taken from [Smith 88]:

$$J_{\ominus}\{\mathbf{x}\} = \begin{pmatrix} -R^T & N \\ O_{3\times 3} & Q \end{pmatrix}$$

where:

$$R = Mrot(\mathbf{x}) = \begin{pmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} \end{pmatrix}$$

$$N = \begin{pmatrix} 0 & -n_z x\cos\phi - n_z y\sin\phi + z\cos\theta & n_y x - n_x y \\ z' & -o_z x\cos\phi - o_z y\sin\phi + z\sin\theta\sin\psi & o_y x - o_x y \\ -y' & -a_z x\cos\phi - a_z y\sin\phi + z\sin\theta\cos\psi & a_y x - a_x y \end{pmatrix}$$

$$Q = \begin{pmatrix} -n_x/(1-a_x^2) & -o_x\cos\psi/(1-a_x^2) & a_z a_x/(1-a_x^2) \\ o_x/(1-a_x^2)^{1/2} & -a_z\cos\phi/(1-a_x^2)^{1/2} & a_y/(1-a_x^2)^{1/2} \\ n_x a_x/(1-a_x^2) & -a_y\cos\phi/(1-a_x^2) & -a_z/(1-a_x^2) \end{pmatrix}$$

where y' and z' are the corresponding components of $\ominus\mathbf{x}$.

In the particular case where $\mathbf{x}$ equals zero, the Jacobian is equal to:

$$J_{\ominus}\{0\} = -I_6$$

A.4 Differential Transformations and Jacobian of a Transformation

Let A and B be two references, whose relative location is given by the location vector $\mathbf{x}_{AB}$. A differential change in the location of reference B which takes it to B' (close to B) can be represented by means of a *differential transformation*[2], that is, a transformation formed by a translation and a rotation of differential values:

$$t = Trasl(dx, dy, dz)\ Rot(z, d\phi_z)\ Rot(y, d\theta_y)\ Rot(x, d\psi_x)$$

in two alternative ways: taking reference A as base reference, or taking reference B. If we represent transformations by means of location vectors, the two alternative expressions are:

$$\begin{aligned} \mathbf{x}_{AB'} &= \mathbf{x}_{AB} \oplus \mathbf{d}_B \\ \mathbf{x}_{AB'} &= \mathbf{d}_A \oplus \mathbf{x}_{AB} \end{aligned}$$

where $\mathbf{d}_A$ y $\mathbf{d}_B$ are two *differential location vectors* (vectors of differential movement of the form $\mathbf{d} = (dx, dy, dz, d\psi_x, d\theta_y, d\phi_z)^T$, in the terminology of [Paul 81]).

Their representation by means of a homogeneous matrix can be obtained from the general expression of $Hom(\mathbf{x})$, taking into account that $cos(d\alpha) \simeq 1$ y $sin(d\alpha) \simeq d\alpha$:

$$Hom(\mathbf{d}) = \begin{pmatrix} 1 & -d\phi_z & d\theta_y & dx \\ d\phi_z & 1 & -d\psi_x & dy \\ -d\theta_y & d\psi_x & 1 & dz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The differential location vectors expressed in references A and B are related by the Jacobian of the relative transformation [Paul 81]:

$$\begin{aligned} \mathbf{d}_A &= J\{\mathbf{x}_{AB}\}\mathbf{d}_B \\ \mathbf{d}_B &= J\{\mathbf{x}_{BA}\}\mathbf{d}_A = J^{-1}\{\mathbf{x}_{AB}\}\mathbf{d}_A \end{aligned}$$

The direct and inverse Jacobians can be calculated as:

[2] Note that [Paul 81] uses the term “differential transformation” is a different sense, to mean a homogeneous matrix dT such that $T_{AB'} = T_{AB} + dT$, where T_{AB} and $T_{AB'}$ are homogeneous matrices representing the relative transformation between A and B, and A and B', respectively.

$$J_{AB} = J\{\mathbf{x}_{AB}\} = \begin{pmatrix} R & S \\ 0_{3\times 3} & R \end{pmatrix}$$

$$J_{BA} = J\{\mathbf{x}_{BA}\} = J^{-1}\{\mathbf{x}_{AB}\} = \begin{pmatrix} R^T & S^T \\ 0_{3\times 3} & R^T \end{pmatrix}$$

where:

$$S = (\mathbf{p} \times \mathbf{n} \quad \mathbf{p} \times \mathbf{o} \quad \mathbf{p} \times \mathbf{a})$$

$$Hom(\mathbf{x}_{AB}) = \begin{pmatrix} R & \mathbf{p} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The relationship between J_{AB} and the Jacobians of the composition of transformations is given by:

$$\begin{aligned} \mathbf{x}_{AB'} &= \mathbf{x}_{AB} \oplus \mathbf{d}_B = \mathbf{x}_{AB} + J_{2\oplus}\{\mathbf{x}_{AB}, 0\}\, \mathbf{d}_B \\ \mathbf{x}_{AB'} &= \mathbf{d}_A \oplus \mathbf{x}_{AB} = \mathbf{x}_{AB} + J_{1\oplus}\{0, \mathbf{x}_{AB}\}\, \mathbf{d}_A \end{aligned}$$

so:

$$\mathbf{d}_A = J_{1\oplus}^{-1}\{0, \mathbf{x}_{AB}\}\;\; J_{2\oplus}\{\mathbf{x}_{AB}, 0\}\, \mathbf{d}_B = J_{AB}\, \mathbf{d}_B$$

and thus:

$$\begin{aligned} J_{AB} &= J_{1\oplus}^{-1}\{0, \mathbf{x}_{AB}\}\;\; J_{2\oplus}\{\mathbf{x}_{AB}, 0\} \\ J_{BA} &= J_{2\oplus}^{-1}\{\mathbf{x}_{AB}, 0\}\;\; J_{1\oplus}\{0, \mathbf{x}_{AB}\} \end{aligned}$$

Appendix B

Aligning Transformations

In this appendix we give the aligning transformations for all combinations of geometric elements, that is, all combinations of subgroups of symmetries. For each case, we describe a solution for the general case. The solution may appear in terms of the components of the relative location vector between the features, or of the elements of the homogeneous matrix corresponding to their relative transformation. The values of the location vector for particular cases where the given solution is not valid are also given.

B.1 $\mathcal{S}_A = R_x, \;\; \mathcal{S}_B = R_x$

Given two features A and B with symmetries $\mathcal{S}_A = R_x$ and $\mathcal{S}_B = R_x$, such as two circles, their aligning references have the following form:

$$\begin{aligned}
\mathbf{x}_{A\bar{A}} &= (0,\ 0,\ 0,\ \psi_a,\ 0,\ 0)^T \\
\mathbf{x}_{B\bar{B}} &= (0,\ 0,\ 0,\ \psi_b,\ 0,\ 0)^T
\end{aligned}$$

In the general case, *three distances* and *one angle* can be measured between them, using the following values for the aligning transformations:

$$\begin{aligned}
\psi_a &= -\tan^{-1}\left(\tfrac{\sin\theta}{\sin\phi\cos\theta}\right) &&= \tan^{-1}\left(\tfrac{n_z}{n_y}\right) \\
\psi_b &= \tan^{-1}\left(\tfrac{\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi}{\cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi}\right) &&= \tan^{-1}\left(\tfrac{a_x}{o_x}\right)
\end{aligned}$$

$$\begin{aligned}
\mathbf{x}_{\bar{A}\bar{B}} &= (\bar{x},\ \bar{y},\ \bar{z},\ 0,\ 0,\ \bar{\phi})^T \\
\bar{x} &= x &&= p_x \\
\bar{y} &= \tfrac{y\,\cos\theta\sin\phi-z\,\sin\theta}{\sqrt{1-\cos^2\theta\cos^2\phi}} &&= \tfrac{p_y\,n_y+p_z\,n_z}{\sqrt{1-n_x^2}} \\
\bar{z} &= \tfrac{z\,\cos\theta\sin\phi+y\,\sin\theta}{\sqrt{1-\cos^2\theta\cos^2\phi}} &&= \tfrac{p_z\,n_y-p_y\,n_z}{\sqrt{1-n_x^2}} \\
\bar{\phi} &= \tan^{-1}\left(\tfrac{\sqrt{1-\cos^2\theta\cos^2\phi}}{\cos\theta\cos\phi}\right) &&= \tan^{-1}\left(\tfrac{\sqrt{1-n_x^2}}{n_x}\right)
\end{aligned}$$

This solution cannot be applied when $\theta = \phi = 0$. In this case, which corresponds to a situation where the x axes of both references are parallel, we can measure *two distances*. The solution is:

$$\begin{aligned}
\psi_a &= \tan^{-1}\left(\tfrac{z}{y}\right) &&= \tan^{-1}\left(\tfrac{p_z}{p_y}\right) \\
\psi_b &= \tan^{-1}\left(\tfrac{z\,\cos\psi-y\,\sin\psi}{z\,\sin\psi+y\,\cos\psi}\right) &&= \tan^{-1}\left(\tfrac{p_z\,a_z-p_y\,o_z}{p_z\,o_z+p_y\,a_z}\right)
\end{aligned}$$

$$\begin{aligned}
\mathbf{x}_{\bar{A}\bar{B}} &= (\bar{x},\ \bar{y},\ 0,\ 0,\ 0,\ 0)^T \\
\bar{x} &= x \\
\bar{y} &= \sqrt{y^2+z^2}
\end{aligned}$$

Again, this solution cannot be applied when $y = z = \theta = \phi = 0$. This case corresponds to a situation where the two x axes of the references are aligned. In this case, there is no unique solution.

B.2 $\mathcal{S}_A = R_x, \;\; \mathcal{S}_B = R_{xyz}$

Given two features A and B with symmetries $\mathcal{S}_A = R_x$ and $\mathcal{S}_B = R_{xyz}$, such as a circle and a vertex, their aligning references have the following form:

$$\begin{array}{rcl} \mathbf{x}_{A\bar{A}} & = & (0,\ 0,\ 0,\ \psi_a,\ 0,\ 0)^T \\ \mathbf{x}_{B\bar{B}} & = & (0,\ 0,\ 0,\ \psi_b,\ \theta_b,\ \phi_b)^T \end{array}$$

In the general case, *two distances* can be measured, using the following values for the aligning transformations:

$$\begin{array}{rclcl} \psi_a & = & -\tan^{-1}\left(\frac{y}{z}\right) & & \\ \psi_b & = & -\tan^{-1}\left(\frac{y\cos\theta\cos\psi - z(\sin\phi\sin\theta\cos\psi + \cos\phi\sin\psi}{y(\cos\psi\sin\phi\sin\theta - \cos\phi\sin\psi) + z\cos\theta\cos\psi}\right) & = & -\tan^{-1}\left(\frac{ya_z - za_y}{ya_y + za_z}\right) \\ \theta_b & = & \tan^{-1}\left(\frac{-\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi}{\sqrt{1-(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)^2}}\right) & = & \tan^{-1}\left(\frac{-a_x}{\sqrt{1-a_x^2}}\right) \\ \phi_b & = & \tan^{-1}\left(\frac{\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi}{\cos\phi\cos\theta}\right) & = & \tan^{-1}\left(\frac{o_x}{n_x}\right) \end{array}$$

$$\begin{array}{rcl} \mathbf{x}_{\bar{A}\bar{B}} & = & (\bar{x},\ 0,\ \bar{z},\ 0,\ 0,\ 0)^T \\ \bar{x} & = & x \\ \bar{z} & = & \sqrt{y^2 + z^2} \end{array}$$

This solution cannot be applied when $y = z = 0$. In this case, which corresponds to a situation where the vertex belongs to the x axis of the circle, the solution is not unique, and thus, the two features cannot be used to determine the location of an object.

B.3 $\mathcal{S}_A = T_x, \;\; \mathcal{S}_B = T_x$

Given two features A and B with symmetries $\mathcal{S}_A = T_x$ and $\mathcal{S}_B = T_x$, such as two dihedrals, their aligning references have the following form:

$$\begin{array}{rcl} \mathbf{x}_{A\bar{A}} & = & (x_a,\ 0,\ 0,\ 0,\ 0,\ 0)^T \\ \mathbf{x}_{B\bar{B}} & = & (x_b,\ 0,\ 0,\ 0,\ 0,\ 0)^T \end{array}$$

In the general case, *one distance* and *three angles* can be measured, using the following values for the aligning transformations:

$$\begin{array}{rcl} x_a & = & \frac{x\ \sin\theta\cos\phi + z\ \cos\theta}{\sin\phi} \\ x_b & = & \frac{z}{\sin\theta} \end{array}$$

$$\begin{array}{rcl} \mathbf{x}_{\bar{A}\bar{B}} & = & (0,\ \bar{y},\ 0,\ \bar{\psi},\ \bar{\theta},\ \bar{\phi})^T \\ \bar{y} & = & y + \frac{\sin\phi\cos\theta}{\sin\theta}\, z \\ \bar{\psi} & = & \psi \\ \bar{\theta} & = & \theta \\ \bar{\phi} & = & \phi \end{array}$$

This solution cannot be applied when $\theta = 0$. In this case, we can measure *one distance* and *two angles*. The solution is:

$$\begin{array}{rcl} x_a & = & x - \frac{\cos\phi}{\sin\phi}\, y \\ x_b & = & -\frac{y}{\sin\phi} \end{array}$$

$$\begin{array}{rcl} \mathbf{x}_{\bar{A}\bar{B}} & = & (0,\ 0,\ \bar{z},\ \bar{\psi},\ 0,\ \bar{\phi})^T \\ \bar{z} & = & z \\ \bar{\psi} & = & \psi \\ \bar{\phi} & = & \phi \end{array}$$

Again, this solution cannot be applied when $\theta = \phi = 0$. This case corresponds to a situation where the two x axes of the references are aligned. In this case, there is no unique solution.

B.4 $\mathcal{S}_A = T_x, \ \mathcal{S}_B = R_x$

Given two features A and B with symmetries $\mathcal{S}_A = T_x$ and $\mathcal{S}_B = R_x$, such as a dihedral and a circle, their aligning references have the following form:

$$\begin{aligned} \mathbf{x}_{A\bar{A}} &= (x_a, \ 0, \ 0, \ 0, \ 0, \ 0)^T \\ \mathbf{x}_{B\bar{B}} &= (0, \ 0, \ 0, \ \psi_b, \ 0, \ 0)^T \end{aligned}$$

In the general case, *two distances* and *two angles* can be measured, using the following values for the aligning transformations:

$$\begin{aligned} x_a &= x \\ \psi_b &= -\psi \end{aligned}$$

$$\begin{aligned} \mathbf{x}_{\bar{A}\bar{B}} &= (0, \ \bar{y}, \ \bar{z}, \ 0, \ \bar{\theta}, \ \bar{\phi})^T \\ \bar{y} &= y \\ \bar{z} &= z \\ \bar{\theta} &= \theta \\ \bar{\phi} &= \phi \end{aligned}$$

This solution is always valid.

B.5 $\mathcal{S}_A = T_x\, R_x,\ \ \mathcal{S}_B = T_x$

Given two features A and B with symmetries $\mathcal{S}_A = T_x\, R_x$ and $\mathcal{S}_B = T_x$, such as an edge and a dihedral, their aligning references have the following form:

$$\begin{aligned} \mathbf{x}_{A\bar{A}} &= (x_a,\ 0,\ 0,\ \psi_a,\ 0,\ 0)^T \\ \mathbf{x}_{B\bar{B}} &= (x_b,\ 0,\ 0,\ 0,\ 0,\ 0)^T \end{aligned}$$

In the general case, *one distance* and *two angles* can be measured, using the following values for the aligning transformations:

$$\begin{array}{rcl@{\qquad}cl}
x_a &=& x_b \cos\theta\cos\phi + x &=& x_b\, n_x + p_x \\
\psi_a &=& \tan^{-1}\left(\frac{-\sin\theta}{\sin\phi\cos\theta}\right) &=& \tan^{-1}\left(\frac{n_z}{n_y}\right) \\
x_b &=& \frac{y\sin\phi\cos\theta - z\sin\theta}{\cos\theta^2\cos\phi^2 - 1} &=& \frac{p_y\, n_y + p_z\, n_z}{n_x^2 - 1} \\
&&&& \\
\mathbf{x}_{\bar{A}\bar{B}} &=& (0,\ 0,\ \bar{z},\ \bar{\psi},\ 0,\ \bar{\phi})^T && \\
\bar{z} &=& \frac{y\sin\theta + z\sin\phi\cos\theta}{\sqrt{1-\cos\theta^2\cos\phi^2}} &=& \frac{(p_z\, n_y - p_y\, n_z)}{\sqrt{1-n_x^2}} \\
\bar{\psi} &=& -\tan^{-1}\left(\frac{\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi}{\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi}\right) &=& \tan^{-1}\left(\frac{a_x}{-o_x}\right) \\
\bar{\phi} &=& \tan^{-1}\left(\frac{\sqrt{1-\cos\theta^2\cos\phi^2}}{\cos\theta\cos\phi}\right) &=& \tan^{-1}\left(\frac{\sqrt{1-n_x^2}}{n_x}\right)
\end{array}$$

This solution is not valid when $\theta = \phi = 0$. In this case, which corresponds to the situation where the edge and the dihedral are parallel, there is no unique solution.

B.6 $\mathcal{S}_A = R_{xyz}, \;\; \mathcal{S}_B = T_x$

Given two features A and B with symmetries $\mathcal{S}_A = R_{xyz}$ and $\mathcal{S}_B = T_x$, such as a vertex and a dihedral, their aligning references have the following form:

$$\begin{aligned} \mathbf{x}_{A\bar{A}} &= (0,\ 0,\ 0,\ \psi_a,\ \theta_a,\ \phi_a)^T \\ \mathbf{x}_{B\bar{B}} &= (x_b,\ 0,\ 0,\ 0,\ 0,\ 0)^T \end{aligned}$$

In the general case, *two distances* can be measured, using the following values for the aligning transformations:

$$\begin{array}{rcll} \psi_a &=& \psi & \\ \theta_a &=& \theta & \\ \phi_a &=& \phi & \\ x_b &=& -x\cos\theta\cos\phi - y\cos\theta\sin\phi + z\sin\theta &= \;\; -\mathbf{p}\cdot\mathbf{n} \\ & & & \\ \mathbf{x}_{\bar{A}\bar{B}} &=& (0,\ \bar{y},\ \bar{z},\ 0,\ 0,\ 0)^T & \\ \bar{y} &=& x\,(\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi) & \\ & & +y\,(\sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi) & \\ & & +z\sin\psi\cos\theta &= \;\; \mathbf{p}\cdot\mathbf{o} \\ \bar{z} &=& x\,(\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi) & \\ & & +y\,(\cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi) & \\ & & +z\cos\psi\cos\theta &= \;\; \mathbf{p}\cdot\mathbf{a} \end{array}$$

This solution is always valid.

B.7 $\mathcal{S}_A = T_x\, R_x,\ \ \mathcal{S}_B = R_x$

Given two features A and B with symmetries $\mathcal{S}_A = T_x\, R_x$ and $\mathcal{S}_B = R_x$, such as an edge and a circle, their aligning references have the following form:

$$\begin{array}{rcl} \mathbf{x}_{A\bar{A}} & = & (x_a,\ 0,\ 0,\ \psi_a,\ 0,\ 0)^T \\ \mathbf{x}_{B\bar{B}} & = & (0,\ 0,\ 0,\ \psi_b,\ 0,\ 0)^T \end{array}$$

In the general case, *two distances* and *one angle* can be measured, using the following values for the aligning transformations:

$$\begin{array}{rclcl} x_a & = & x & & \\ \psi_a & = & \tan^{-1}\left(\frac{-\sin\theta}{\sin\phi\cos\theta}\right) & = & \tan^{-1}\left(\frac{n_z}{n_y}\right) \\ \psi_b & = & \tan^{-1}\left(\frac{\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi}{\cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi}\right) & = & \tan^{-1}\left(\frac{a_x}{o_x}\right) \end{array}$$

$$\begin{array}{rclcl} \mathbf{x}_{\bar{A}\bar{B}} & = & (0,\ \bar{y},\ \bar{z},\ 0,\ 0,\ \bar{\phi})^T & & \\ \bar{y} & = & \frac{y\sin\phi\cos\theta-z\sin\theta}{\sqrt{1-\cos\theta^2\cos^2\phi}} & = & \frac{p_y\,n_y+p_z\,n_z}{\sqrt{1-n_x^2}} \\ \bar{z} & = & \frac{y\sin\theta+z\sin\phi\cos\theta}{\sqrt{1-\cos\theta^2\cos\phi^2}} & = & \frac{p_z\,n_y-p_y\,n_z}{\sqrt{1-n_x^2}} \\ \bar{\phi} & = & \tan^{-1}\left(\frac{\sqrt{1-\cos\theta^2\cos^2\phi}}{\cos\theta\cos\phi}\right) & = & \tan^{-1}\left(\frac{\sqrt{1-n_x^2}}{n_x}\right) \end{array}$$

This solution not valid when $\theta = \phi = 0$. In this case, which corresponds to a situation where the edge is parallel to the normal of the circle, there is no unique solution.

B.8 $\mathcal{S}_A = T_x\, R_x,\ \ \mathcal{S}_B = T_x\, R_x$

Given two features A and B with symmetries $\mathcal{S}_A = T_x\, R_x$ and $\mathcal{S}_B = T_x\, R_x$, such as two edges, their aligning references have the following form:

$$\begin{array}{rcl}
\mathbf{x}_{A\bar{A}} & = & (x_a,\ 0,\ 0,\ \psi_a,\ 0,\ 0)^T \\
\mathbf{x}_{B\bar{B}} & = & (x_b,\ 0,\ 0,\ \psi_b,\ 0,\ 0)^T
\end{array}$$

In the general case, *one distance* and *one angle* can be measured, using the following values for the aligning transformations:

$$\begin{array}{rclcl}
x_a & = & x - \frac{\cos\phi\cos\theta(y\sin\phi\cos\theta - z\sin\theta)}{1-\cos\phi^2\cos\theta^2} & = & x - \frac{n_x(yn_y+zn_z)}{1-n_x^2} \\
\psi_a & = & -\tan^{-1}\left(\frac{\sin\theta}{\sin\phi\cos\theta}\right) & = & \tan^{-1}\left(\frac{n_z}{n_y}\right) \\
x_b & = & -\frac{y\sin\phi\cos\theta - z\sin\theta}{1-\cos\phi^2\cos\theta^2} & = & -\frac{yn_y+zn_z}{1-n_x^2} \\
\psi_b & = & \tan^{-1}\left(\frac{\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi}{\cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi}\right) & = & \tan^{-1}\left(\frac{a_x}{o_x}\right)
\end{array}$$

$$\begin{array}{rclcl}
\mathbf{x}_{\bar{A}\bar{B}} & = & (0,\ 0,\ \bar{z},\ 0,\ 0,\ \bar{\phi})^T & & \\
\bar{z} & = & \frac{z\sin\phi\cos\theta+y\sin\theta}{\sqrt{1-\cos\phi^2\cos\theta^2}} & = & \frac{zn_y-yn_z}{\sqrt{1-n_x^2}} \\
\bar{\phi} & = & \tan^{-1}\left(\frac{\sqrt{1-\cos\phi^2\cos\theta^2}}{\cos\phi\cos\theta}\right) & = & \tan^{-1}\left(\frac{\sqrt{1-n_x^2}}{n_x}\right)
\end{array}$$

This solution not valid when $\theta = \phi = 0$. In this case, which corresponds to a situation where the edges are parallel, there is no unique solution.

B.9 $\mathcal{S}_A = T_x\, R_x, \;\; \mathcal{S}_B = R_{xyz}$

Given two features A and B with symmetries $\mathcal{S}_A = T_x\, R_x$ and $\mathcal{S}_B = R_{xyz}$, such as an edge and a vertex, their aligning references have the following form:

$$\begin{array}{rcl}
\mathbf{x}_{A\bar{A}} & = & (x_a,\ 0,\ 0,\ \psi_a,\ 0,\ 0)^T \\
\mathbf{x}_{B\bar{B}} & = & (0,\ 0,\ 0,\ \psi_b,\ \theta_b,\ \phi_b)^T
\end{array}$$

In the general case, one distance can be measured, using the following values for the aligning transformations:

$$\begin{array}{rclcl}
x_a & = & x & & \\
\psi_a & = & -\tan^{-1}\left(\frac{y}{z}\right) & & \\
\psi_b & = & -\tan^{-1}\left(\frac{y\cos\theta\cos\psi - z(\sin\phi\sin\theta\cos\psi + \cos\phi\sin\psi}{y(\cos\psi\sin\phi\sin\theta - \cos\phi\sin\psi) + z\cos\theta\cos\psi}\right) & = & -\tan^{-1}\left(\frac{ya_z - za_y}{ya_y + za_z}\right) \\
\theta_b & = & \tan^{-1}\left(\frac{-\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi}{\sqrt{1 - (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)^2}}\right) & = & \tan^{-1}\left(\frac{-a_x}{\sqrt{1 - a_x^2}}\right) \\
\phi_b & = & \tan^{-1}\left(\frac{\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi}{\cos\phi\cos\theta}\right) & = & \tan^{-1}\left(\frac{o_x}{n_x}\right)
\end{array}$$

$$\begin{array}{rcl}
\mathbf{x}_{\bar{A}\bar{B}} & = & (0,\ 0,\ \bar{z},\ 0,\ 0,\ 0)^T \\
\bar{z} & = & \sqrt{y^2 + z^2}
\end{array}$$

This solution cannot be applied when $y = z = 0$. In this case, which corresponds to a situation where the vertex belongs to the edge, the solution is not unique, and thus, the two features cannot be used to determine the location of an object.

B.10 $\mathcal{S}_A = T_{xy}\, R_z,\ \ \mathcal{S}_B = R_x$

Given two features A and B with symmetries $\mathcal{S}_A = T_{xy}\, R_z$ and $\mathcal{S}_B = R_x$, such as a plane and a circle, their aligning references have the following form:

$$\begin{aligned}
\mathbf{x}_{A\bar{A}} &= (x_a,\ y_a,\ 0,\ 0,\ 0,\ \phi_a)^T \\
\mathbf{x}_{B\bar{B}} &= (0,\ 0,\ 0,\ \psi_b,\ 0,\ 0)^T
\end{aligned}$$

In the general case, *one distance* and *one angle* can be measured, using the following values for the aligning transformations:

$$\begin{aligned}
x_a &= x \\
y_a &= y \\
\phi_a &= \phi \\
\psi_b &= -\psi
\end{aligned}$$

$$\begin{aligned}
\mathbf{x}_{\bar{A}\bar{B}} &= (0,\ 0,\ \bar{z},\ 0,\ \bar{\theta},\ 0)^T \\
\bar{z} &= z \\
\bar{\theta} &= \theta
\end{aligned}$$

This solution is always valid.

B.11 $\mathcal{S}_A = T_{xy}\,R_z,\;\; \mathcal{S}_B = R_{xyz}$

Given two features A and B with symmetries $\mathcal{S}_A = T_{xy}\,R_z$ and $\mathcal{S}_B = R_{xyz}$, such as a plane and a vertex, their aligning references have the following form:

$$\begin{array}{rcl} \mathbf{x}_{A\bar{A}} & = & (x_a,\ y_a,\ 0,\ 0,\ 0,\ \phi_a)^T \\ \mathbf{x}_{B\bar{B}} & = & (0,\ 0,\ 0,\ \psi_b,\ \theta_b,\ \phi_b)^T \end{array}$$

In the general case, *one distance* can be measured, using the following values for the aligning transformations:

$$\begin{array}{rcll} x_a & = & x & \\ y_a & = & y & \\ \phi_a & = & \text{free} & \\ \psi_b & = & \tan^{-1}\left(\frac{\sin\phi\sin\theta\cos\psi-\cos\phi\sin\psi}{\cos\theta\cos\psi}\right) & = \tan^{-1}\left(\frac{a_y}{a_z}\right) \\ \theta_b & = & -\tan^{-1}\left(\frac{\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi}{\sqrt{\cos^2\phi\cos^2\theta-(\cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi)^2}}\right) & \\ & = & -\tan^{-1}\left(\frac{a_x}{\sqrt{n_x^2+o_x^2}}\right) & \\ \phi_b & = & \tan^{-1}\left(\frac{\cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi}{\cos\phi\cos\theta}\right) & = \tan^{-1}\left(\frac{o_x}{n_x}\right) \\ \mathbf{x}_{\bar{A}\bar{B}} & = & (0,\ 0,\ \bar{z},\ 0,\ 0,\ 0)^T & \\ \bar{z} & = & z & \end{array}$$

Note that ϕ_a remains free. This means that there is no unique solution, and thus, two features of these types are not sufficient to completely determine the location of an object. This solution is always valid.

B.12 $\mathcal{S}_A = T_{xy}\,R_z,\ \ \mathcal{S}_B = T_x$

Given two features A and B with symmetries $\mathcal{S}_A = T_{xy}\,R_z$ and $\mathcal{S}_B = T_x$, such as a plane and a dihedral, their aligning references have the following form:

$$\begin{aligned}
\mathbf{x}_{A\bar{A}} &= (x_a,\ y_a,\ 0,\ 0,\ 0,\ \phi_a)^T \\
\mathbf{x}_{B\bar{B}} &= (x_b,\ 0,\ 0,\ 0,\ 0,\ 0)^T
\end{aligned}$$

In the general case, *two angles* can be measured, using the following values for the aligning transformations:

$$\begin{aligned}
x_a &= x + z\,\tfrac{\cos\phi\cos\theta}{\sin\theta} \\
y_a &= y + z\,\tfrac{\sin\phi\cos\theta}{\sin\theta} \\
\phi_a &= \phi \\
x_b &= \tfrac{z}{\sin\theta}
\end{aligned}$$

$$\begin{aligned}
\mathbf{x}_{\bar{A}\bar{B}} &= (0,\ 0,\ 0,\ \bar{\psi},\ \bar{\theta},\ 0)^T \\
\bar{\psi} &= \psi \\
\bar{\theta} &= \theta
\end{aligned}$$

This solution is not valid when $\theta = 0$. In this case there is no unique solution.

B.13 $\mathcal{S}_A = T_{xy}\,R_z,\ \ \mathcal{S}_B = T_x\,R_x$

Given two features A and B with symmetries $\mathcal{S}_A = T_{xy}\,R_z$ and $\mathcal{S}_B = T_x\,R_x$, such as a plane and an edge, their aligning references have the following form:

$$\begin{aligned}
\mathbf{x}_{A\bar{A}} &= (x_a,\ y_a,\ 0,\ 0,\ 0,\ \phi_a)^T \\
\mathbf{x}_{B\bar{B}} &= (x_b,\ 0,\ 0,\ \psi_b,\ 0,\ 0)^T
\end{aligned}$$

In the general case, *one angle* can be measured, using the following values for the aligning transformations:

$$\begin{aligned}
x_a &= x + z\,\tfrac{\cos\phi\cos\theta}{\sin\theta} \\
y_a &= y + z\,\tfrac{\sin\phi\cos\theta}{\sin\theta} \\
\phi_a &= \phi \\
x_b &= \tfrac{z}{\sin\theta} \\
\psi_b &= -\psi
\end{aligned}$$

$$\begin{aligned}
\mathbf{x}_{\bar{A}\bar{B}} &= (0,\ 0,\ 0,\ 0,\ \bar{\theta},\ 0)^T \\
\bar{\theta} &= \theta
\end{aligned}$$

This solution is not valid when $\theta = 0$. In this case, which corresponds to a situation where the plane and the edge are parallel, there is no unique solution.

B.14 $\mathcal{S}_A = T_{xy}\, R_z,\ \ \mathcal{S}_B = T_{xy}\, R_z$

Given two features A and B with symmetries $\mathcal{S}_A = T_{xy}\, R_z$ and $\mathcal{S}_B = T_{xy}\, R_z$, such as two planes, their aligning references have the following form:

$$\begin{array}{rcl} \mathbf{x}_{A\bar{A}} & = & (x_a,\ y_a,\ 0,\ 0,\ 0,\ \phi_a)^T \\ \mathbf{x}_{B\bar{B}} & = & (x_b,\ y_b,\ 0,\ 0,\ 0,\ \phi_b)^T \end{array}$$

In the general case, one angle can be measured, using the following values for the aligning transformations:

$$\begin{array}{rclcl} x_a & = & x + \frac{(y-y_a)(\sin\theta\sin\phi\cos\psi - \sin\psi\cos\phi) + z\cos\theta\cos\psi}{\sin\theta\cos\phi\cos\psi + \sin\psi\sin\phi} & = & x + \frac{(y-y_a)a_y + za_z}{a_x} \\ y_a & & \text{free} & & \\ \phi_a & = & \tan^{-1}\left(\frac{\sin\theta\sin\phi\cos\psi - \sin\psi\cos\phi}{\sin\theta\cos\phi\cos\psi + \sin\psi\sin\phi}\right) & = & \tan^{-1}\left(\frac{a_y}{a_x}\right) \\ x_b & = & -\frac{(y-y_a)\cos\theta\sin\psi + z(-\cos\phi\cos\psi - \sin\theta\sin\psi\sin\phi)}{\sin\theta\cos\phi\cos\psi + \sin\psi\sin\phi} & = & -\frac{(y-y_a)o_z + zo_y}{a_x} \\ y_b & = & -\frac{z\sin\phi\cos\theta + (y-y_a)\sin\theta}{\sin\theta\cos\phi\cos\psi + \sin\psi\sin\phi} & = & -\frac{zn_y - (y-y_a)n_z}{a_x} \\ \phi_b & = & -\tan^{-1}\left(\frac{\cos\theta\sin\psi}{\sin\theta}\right) & = & \tan^{-1}\left(\frac{n_y}{n_z}\right) \end{array}$$

$$\begin{array}{rclcl} \mathbf{x}_{\bar{A}\bar{B}} & = & (0,\ 0,\ 0,\ 0,\ \bar{\theta},\ 0)^T & & \\ \bar{\theta} & = & -\tan^{-1}\left(\frac{\sqrt{1-\cos^2\theta\cos^2\psi}}{\cos\theta\cos\psi}\right) & = & -\tan^{-1}\left(\frac{\sqrt{1-n_x^2}}{n_x^2}\right) \end{array}$$

No the that variable y_a remains free. This means that this solution is not unique, because, two planes are not sufficient to completely determine the location of an object. This solution is not valid when $\theta = \psi = 0$. This corresponds to a situation where the two panes are parallel.

B.15 $\mathcal{S}_A = R_{xyz}$, $\mathcal{S}_B = R_{xyz}$

Given two features A and B with symmetries $\mathcal{S}_A = R_{xyz}$ and $\mathcal{S}_B = R_{xyz}$, such as two vertices, their aligning references have the following form:

$$\begin{array}{rcl} \mathbf{x}_{A\bar{A}} & = & (0,\ 0,\ 0,\ \psi_a,\ \theta_a,\ \phi_a)^T \\ \mathbf{x}_{B\bar{B}} & = & (0,\ 0,\ 0,\ \psi_b,\ \theta_b,\ \phi_b)^T \end{array}$$

In the general case, *one distance* can be measured, using the following values for the aligning transformations:

$$\begin{array}{rcl} \psi_a & = & 0 \\ \theta_a & = & \tan^{-1}\left(\frac{x}{z}\right) \\ \phi_a & = & -\tan^{-1}\left(\frac{y}{\sqrt{x^2+z^2}}\right) \\ \psi_b & = & \tan^{-1}\left(\frac{(x^2+z^2)a_y+y(xa_x-za_z)}{\sqrt{x^2+z^2}(xa_x+ya_y+za_z)}\right) \\ \theta_b & = & -\tan^{-1}\left(\frac{za_x-xa_z}{(x^2+z^2)-(xa_x+za_z)^2}\right) \\ \phi_b & = & -\tan^{-1}\left(\frac{xo_z+zo_x}{xn_z+zn_x}\right) \end{array}$$

$$\begin{array}{rcl} \mathbf{x}_{\bar{A}\bar{B}} & = & (\bar{x},\ 0,\ 0,\ 0,\ 0,\ 0)^T \\ \bar{x} & = & \sqrt{x^2+y^2+z^2} \end{array}$$

Variable ψ_a remains free. (We have chosen $\psi_a = 0$ for simplicity). This means that this solution is not unique, because two vertices are not sufficient to completely determine the location of an object. This solution is not valid when $x = y = z = 0$. This corresponds to a situation where the two vertices coincide.

Bibliography

[Allen 87] P.K. Allen. *Robotic Object Recognition Using Vision and Touch.* Kluwer Academic Pub., Massachusetts, 1987.

[Bhandarkar 92] S.M. Bhandarkar and M. Suk. Qualitative features and the generalized hough transform. *Pattern Recognition*, 25(9):987–1005, 1992.

[Bolles 82] R.C. Bolles and R.A. Cain. Recognizing and locating partially visible objects: The local-feature-focus method. *Int. J. Robotics Research*, 1(3):57–82, 1982.

[Bolles 86] R.C. Bolles and P. Horaud. 3DPO: A three-dimensional part orientation system. *Int. J. Robotics Research*, 5(2):3–26, 1986.

[Bozzo 83] C.A. Bozzo. *Le Filtrage Optimal et ses Applications aux Problèmes de Poursuite.* Technique et Documentation, Montrouge, France, 1983.

[Brooks 82] R.A. Brooks. Symbolic error analysis and robot planning. *Int. J. Robotics Research*, 1(4):29, 1982.

[Brooks 85] R.A. Brooks. Visual map making for a mobile robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 824–829, St. Louis, Missouri, 1985.

[Cameron 90] A. Cameron and H. Durrant-Whyte. A bayesian approach to optimal sensor placement. *The International Journal of Robotics Research*, 9(5):70–88, 1990.

[Chatila 85] R. Chatila and J. P. Laumond. Position referencing and consistent world modelling for mobile robots. In

IEEE Int. Conf. on Robotics and Automation, pages 138–145, St. Louis, Missouri, 1985.

[Chen 90] C.H. Chen and P. G. Mulgaoukar. Feature utility measures for automatic vision programming. In *IEEE Int. Conf. on Robotics and Automation*, pages 1840–1845, 1990.

[Chiou 92] R.N. Chiou, K.C. Hung, J.K. Guo, C.H. Chen, T.I. Fan, and J.Y. Lee. Polyhedron recognition using three-view analysis. *Pattern Recognition*, 25(1):1–15, 1992.

[Crowley 87] J.L. Crowley. Using the composite surface model for perceptual tasks. In *IEEE Int. Conf. on Robotics and Automation*, pages 929–934, Raleigh, North Carolina, 1987.

[Cuadras 89] C.M. Cuadras. Distancias estadísticas. *Estadística Española*, 30(119):295–378, 1989.

[Durrant-Whyte 87] H.F. Durrant-Whyte. Consistent integration and propagation of disparate sensor observations. *The International Journal of Robotics Research*, 6(3):3–24, 1987.

[Durrant-Whyte 88] H.F. Durrant-Whyte. *Integration, Coordination and Control of Multi-Sensor Robot Systems.* Kluwer Academic Pub., Massachusetts, 1988.

[Ellis 91] R.E. Ellis. Geometric uncertainty in polyhedral object recognition. *IEEE Transactions on Robotics and Automation*, 7(3):361–371, June 1991.

[Ellis 92] R.E. Ellis. Planning tactile recognition paths in two and three dimensions. *The International Journal of Robotics Research*, 11(2):87–111, 1992.

[Faugeras 86] O.D. Faugeras and M. Hebert. The representation recognition, and locating of 3D objects. *Int. J. Robotics Research*, 5(3):27–52, 1986.

[Fisher 91] R.B. Fisher and M.J.L. Orr. Geometric reasoning in a parallel network. *The International Journal of Robotics Research*, 10(2):103–122, 1991.

[Franklin 68] J.N. Franklin. *Matrix Theory.* Prentice-Hall, Englewoods Cliffs, New Jersey, 1968.

[Gelb 74] A. Gelb, editor. *Applied Optimal Estimation.* The M.I.T. Press, Cambridge, Mass., 1974.

[Granjean 91] P. Granjean, M. Ghallab, and E. Dernneuvel. Multisensory scene interpretation: Model-based object recognition. In *IEEE International Conference on Robotics and Automation*, pages 2:352–2:358, 1991.

[Grimson 84] W.E.L. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range or tactile data. *Int. J. Robotics Research*, 3(3):3–35, 1984.

[Grimson 86] W.E.L. Grimson. Sensing strategies for disambiguating among multiple objects in known poses. *IEEE Journal of Robotics and Automation*, 2(4):196–213, 1986.

[Grimson 87] W.E.L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(4):469–482, 1987.

[Grimson 90a] W.E.L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* The MIT Press, Massachusetts, 1990.

[Grimson 90b] W.E.L. Grimson and D. Huttenlocher. On the verification of hypotesized matches in model-based recognition. In O. Faugueras, editor, *1st European Conference on Computer Vision*, pages 489–498. Springer-Verlag, 1990.

[Hager 93] G.D. Hager, S.P. Engelson, and S. Atiya. On comparing statistical and set-based methods in sensor data fusion. In *IEEE International Conference on Robotics and Automation*, pages 2:352–2:358, Atlanta, USA, May 1993.

[Hall 76] M. Hall. *Theory of Groups.* Chelsea Publishing Company, New York, N.Y., 1976.

[Herranz 91] J. Herranz, J. D. Tardós, and J. Neira. Análisis e integración de un sistema de visión 2D basado en descriptores. In 2^o *Congreso de la Asociación Española de Robótica*, Zaragoza-Spain, Noviembre 1991.

[Hutchinson 88] S.A. Hutchinson, R. L. Cromwell, and A.C. Kak. Planning sensing strategies in a robot work cell with multisensor capabilities. In *IEEE Int. Conference on Robotics and Automation*, pages 282–287, Philadelphia, Pennsylvania, 1988.

[Hutchinson 89] S.A. Hutchinson, R.L. Cromwell, and A.C. Kak. Applying uncertainty reasoning to model based object recognition. In *IEEE Int. Conf. on Robotics and Automation*, pages 541–548, Scottsdale, Arizona, 1989.

[Lacroix 92] S. Lacroix, P. Grandjean, and m. Ghallab. Perception planning for a multi-sensory interpretation machine. In *IEEE International Conference on Robotics and Automation*, pages 1818–1824, Nice, France, 1992.

[Lee 90] S. Lee and H.S. Hahn. An optimal sensory strategy of a proximity sensor system for recognition and localization of polyhedral objects. In *IEEE Int. Conf. on Robotics and Automation*, pages 1666–1671, 1990.

[Lozano-Pérez 87] T. Lozano-Pérez, J.L. Jones, E. Mazer, P.A. O'Donnel, and W.E.L. Grimson. Handey: A robot system that recognizes, plans and manipulates. In *IEEE Int. Conf. on Robotics and Automation*, pages 843–849, Raleigh, North Carolina, 1987.

[Maver 93] J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):417–432, 1993.

[Montano 91] L. Montano and C. Sagüés. Non-contact compliant robot motions: Dynamic behaviour and application to feature localization. In *IMACS-MCTS Symposium*, pages 457–464, Lille, France, May 1991.

[Nagata 91] T. Nagata and H.n Zha. Recognizing and locating a known object from multiple images. *IEEE Transactions on Robotics and Automation*, 7(4):434–448, 1991.

[Nakamura 89] Y. Nakamura and Y. Xu. Geometrical fusion method for multi-sensor robotic systems. In *IEEE Int. Conf. on Robotics and Automation*, pages 668–673, Scottsdale, Arizona, 1989.

[Neira 93a] J. Neira, L. Montano, and J.D. Tardós. Constrant-based object recogintion in multisensor systems. In *IEEE International Conference on Robotics and Automation*, pages 3:135–3:142, Atlanta, USA, May 1993.

[Neira 93b] J. Neira, J.D. Tardós, and L. Montano. Goal-directed perception in multisensor object recognition systems. In *1993 IEEE International Conference on Intelligent Control*, Chicago, USA, August 1993. To Appear.

[Neira 93c] J. Neira, J.D. Tardós, and L. Montano. Selective attention in multisensor object recognition. In *1993 IEEE International Conference on Systems, Man and Cybernetics*, Le Touquet-France, October 1993. To Appear.

[Paul 81] R.P. Paul. *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press, Cambridge, Mass., 1981.

[Pollard 89] S.B. Pollard, T.P. Pridmore, J. Porrill, J.E.W. Mayeheb, and J.P. Frisby. Geometrical modelling from multiple stereo views. *Int. J. Robotics Research*, 8(4):3–32, 1989.

[Popplestone 84] R.J. Popplestone. Group theory and robotics. In M. Brady and R. Paul, editors, *Robotics Research: The First International Symposium*, pages 55–64. MIT Press, 1984.

[Porrill 88] J. Porrill. Optimal combination and constraints for geometrical sensor data. *Int. J. Robotics Research*, 7(6):66–77, 1988.

[Ríos 77] S. Ríos. *Métodos Estadísticos.* Ediciones del Castillo S.A., Madrid, Spain, 1977.

[Sabater 91] A. Sabater and F. Thomas. Set membership approach to the propagation of uncertain geometric information. In *IEEE Int. Conf. on Robotics and Automation*, pages 2718–2723, Sacramento, California, 1991.

[Sagüés 92a] C. Sagüés. *Percepción Activa en Reconocimiento Geométrico Basado en Restriccciones.* PhD thesis, Dpto. de Ingeniería Eléctrica e Informática, University of Zaragoza, Spain, Enero 1992.

[Sagüés 92b] C. Sagüés and L. Montano. Active sensing strategies with non-contact compliant motions for constraint-based recognition. In I. Troch, K. Desoyer, and P. Kopacek, editors, *ROBOT CONTROL 1991 (SYROCO'91)*, pages 475–480. Pergamon Press for the IFAC, 1992.

[Smith 88] 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.

[Tang 92] Y.C. Tang and C.S. George Lee. Optimal strategic recognition of objects based on candidate discriminating graph with coordinated sensors. *IEEE Trans. Systems Man and Cybernetics*, 22(4):647–661, 1992.

[Tardós 91] 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.

[Tardós 92a] J.D. Tardós. Representing partial and uncertain sensorial information using the theory of symmetries. In *IEEE International Conference on Robotics and Automation*, pages 1799–1804, Nice, France, May 1992.

[Tardós 92b] J.D. Tardós, J. Neira, and L. Montano. Multisensor object recognition under uncertainty. In *23rd International Symposium On Industrial Robots*, Barcelona, Spain, October 1992.

[Thomas 88a] F. Thomas. *Planificación de Tareas Robotizadas de Ensamblaje Basada en Análisis de Restricciones.* PhD thesis, Univ. Politecnica de Catalunya, Spain, Junio 1988.

[Thomas 88b] F. Thomas and C. Torras. A group-theoretic approach to the computation of symbolic part relations. *IEEE Journal of Robotics and Automation*, 4(6):622–634, 1988.