Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
C/ María de Luna num. 1
E-50018 Zaragoza
Spain

Internal Report: 1991-03

Active Sensing Using Proximity Sensors for Object Recognition and Localization[1]

Montano L., Sagüés C.

If you want to cite this report, please use the following reference instead:
Active Sensing Using Proximity Sensors for Object Recognition and Localization. Montano L., Sagüés C., *IEEE Fifth International Conference on Advanced Robotics (ICAR91)*, pages 49-54, June 1991.

[1]This work was partially supported by project PA86-0028 of the Comisión Interministerial de Ciencia y Tecnología (CICYT) and by project IT-5/90 of the Diputación General de Aragón (CONAI).

ACTIVE SENSING USING PROXIMITY SENSORS FOR OBJECT RECOGNITION AND LOCALIZATION

MONTANO Luis, SAGÜÉS Carlos

Dpto. de Ingeniería Eléctrica e Informática
Centro Politécnico Superior, Universidad de Zaragoza
María de Luna 3, E-50015 ZARAGOZA, SPAIN

Abstract. In a recognition and localization process, active sensing techniques allow to exploit the discriminant power of geometric features, although they are partially occluded to some sensors. We propose the use of proximity sensors on robot hand and non-contact guarded and compliant motions to sense these features. To carry out recognition, we compute geometric constraints by explicitly considering uncertainty in the acquired features.

1 Introduction

A robot works with objects in a real, prone to uncertainty, environment. To the end of reducing the engineering which supplies the objects in prefixed locations, the robot must be provided with sensorial capabilities. The multisensorial system needs a mechanism to fuse the information acquired, which has an inherent uncertainty. Basically, two approaches have been proposed to model it: some are based on maximal ranks of error [1]; others deal with probabilistic models [2], [3], [4]. We choose a probabilistic model to express the uncertain geometric features acquired by the sensors. In §2 the representation of geometric features is presented.

By using mobile sensors, the system performance is improved. We consider a 2D vision system and mobile proximity sensors on the robot hand. The interest of using proximity sensors comes out of its constructive simplicity, low cost and low computational load to process the information sensed. To drive the proximity sensors, active sensing strategies must be implemented [5], [6]. The main advantage of these techniques appears because they can optimize the number of observations needed to identify and locate objects; in this sense we exploit the discriminant power of some features, although they are partially occluded. In §3 we briefly outline the method to acquire individual features with proximity sensors. We acquire and estimate the location of geometric features

of polyhedral objects, to the end of recognizing and locating them.

A constraint-based recognition [7] may reduce the computational cost of the process. In this paper, we focus the attention in computing some constraints from the sensed features, taking into account the involved uncertainty. In §4 the geometric constraints proposed to be used in the recognition process are stated and computed from the features acquired.

In §5 the method to match estimated and modeled constraints used when recognizing is described.

2 Representing 3-D object geometric features

In this paper we deal with polyhedral objects. We consider them to be built by vertexes, edges and planar surfaces. As told in the introduction, some different approaches have been proposed to represent the features with their location uncertainty. We use a frame attached to each feature [4] expressing its location in a world reference. In Figure 1 the features and the frames are drawn. This is an homogeneous representation for all these features.

In the sequel we express the generic location of frame F in the world reference W by the location vector ${}^W\mathbf{x}_F = (p_x, p_y, p_z, \psi, \theta, \phi)^T$, where (p_x, p_y, p_z) is the origin of the frame and (ψ, θ, ϕ) are the orientation parameters. We choose the Yaw-Pitch-Roll as the orientation angles.

A *vertex* is completely defined by its position: the origin of the frame (p_x, p_y, p_z). The three orientation parameters are degrees of freedom (d.o.f.). An *edge* is represented by a frame with the x axis fitting its direction. The two d.o.f. of an edge in the space are related to one along x axis of the frame and the other to the rotation around it. Additionally, we have a symmetry represented by a 180^o rotation around the z or y axis of the frame. The location of a *plane* is represented by a frame whose origin is on the plane and its z axis is normal to its external face. The three d.o.f. in the space are along the x and y axes and around the z axis. As other representations proposed in the literature, this is overparametrized. In the integration and recognition process we must remove those parameters associated to d.o.f., because they have not valuable information. In [4] a technique based on selection matrices to take the parameters just needed is proposed.

To represent the location uncertainty we use an incremental transformation ${}^F\mathbf{e} \triangleq (\delta p_x, \delta p_y, \delta p_z, \delta\psi, \delta\theta, \delta\phi)^T$ associated to the feature frame F. The true frame location is obtained as:

$$ {}^W\mathbf{x}_F = {}^W\hat{\mathbf{x}}_F \oplus {}^F\mathbf{e} = {}^W\hat{\mathbf{x}}_F \oplus {}^F J_W \, {}^W\mathbf{e} = {}^W J_F \, {}^F\mathbf{e} \oplus {}^W\hat{\mathbf{x}}_F $$

where ${}^W\hat{\mathbf{x}}_F \triangleq E\{{}^W\mathbf{x}_F\} = (\hat{p}_x, \hat{p}_y, \hat{p}_z, \hat{\psi}, \hat{\theta}, \hat{\phi})^T$ is the estimated value of ${}^W\mathbf{x}_F$, ${}^F\mathbf{e}$ and ${}^W\mathbf{e}$ are incremental transformations in F and W frames, ${}^F J_W$ and ${}^W J_F$ the jacobians to relate differential transformations [8], and $\oplus$ is the operator to compose transformations when they are represented as location vectors [3]. Assuming the hypothesis of Gaussian whyte noise, the location uncertainty is

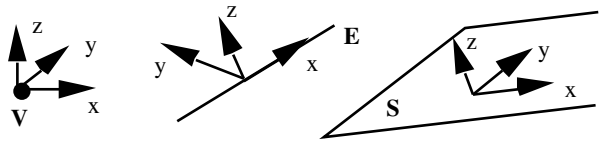

Figure 1: Geometric features and attached frames : (a) vertex V, (b) edge E, (c) planar surface S

characterized by its estimated value, ${}^{F}\hat{\mathbf{e}} = E\{{}^{F}\mathbf{e}\} = 0$ and its covariance matrix $Cov({}^{F}\mathbf{e})$. Then, ${}^{W}\mathbf{x}_F$ is completely defined by ${}^{W}\hat{\mathbf{x}}_F$ and $Cov({}^{F}\mathbf{e})$.

Some interesting advantages of the representation proposed are: (1) uncertainty is represented in a local frame to the geometric feature and so, it is invariant with changes in the reference system; (2) values of the elements of the covariance matrix directly show the significance of errors, (without magnification by the distance to the reference system); and (3), as the estimate errors are small, the covariance matrix is well defined, far from singularities.

3 Estimating location of features with proximity sensors

In this work we model the proximity sensors taking into account measurement device (${}^{S}\mathbf{e}_d$) and robot positioning errors (${}^{R}\mathbf{e}_r$). Our sensorial system has six infrared proximity sensors on the robot gripper (Figure 2). Here, the two frontal sensors are considered. The model is intimately tied to sensor layout on the gripper. It relates the d measurement with the frame location of a point (F).

If we choose the location of the feature in the robot frame (${}^{R}\mathbf{x}_F$) as measurement vector and the location of the feature in the world frame (${}^{W}\mathbf{x}_F$) as state vector, the measurement equation is:

$$\begin{aligned} {}^{R}\mathbf{x}_F &= {}^{R}\hat{\mathbf{x}}_W \oplus {}^{W}\mathbf{x}_F = {}^{R}\hat{\mathbf{x}}_W \oplus ({}^{W}\hat{\mathbf{x}}_F \oplus {}^{F}\mathbf{e}) = \\ &= ({}^{R}\hat{\mathbf{x}}_W \oplus {}^{W}\hat{\mathbf{x}}_F) \oplus {}^{F}\mathbf{e} \end{aligned} \tag{1}$$

As ${}^{R}\mathbf{e}_r$ and ${}^{S}\mathbf{e}_d$ are differencial errors, we can compute the sensor error (${}^{F}\mathbf{e}$) as:

$${}^{F}\mathbf{e} = {}^{F}J_R \, {}^{R}\mathbf{e}_r + {}^{F}J_S \, {}^{S}\mathbf{e}_d \tag{2}$$

where ${}^{F}J_R$ and ${}^{F}J_S$ are the jacobians to relate differential transformations between frames [8].

Our proximity sensors cooperate with a fixed 2D vision system. Some features are not observable by a vision system, but they can be explored by proximity sensors carried by the robot end-effector. We propose data driven strategies

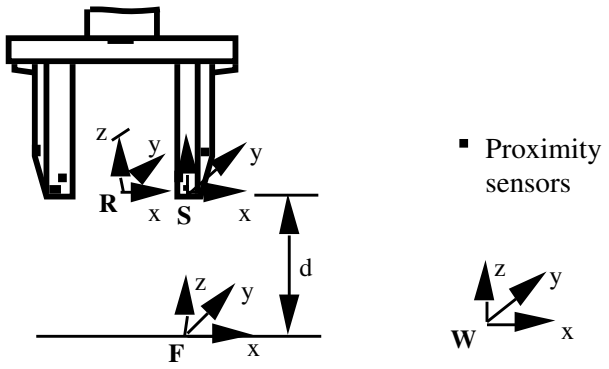

Figure 2: Proximity sensors on the robot gripper and involved frames

for features acquisition with proximity sensors. We consider that we have previous data about the object contours from the vision system. This information is used to drive the robot towards the feature we are going to observe.

The features sensed using the two frontal sensors are edges and planar surfaces. In both, the state to be acquired is the location in the world reference of a frame attached to it, ${}^{W}\mathbf{x}_{F} = {}^{W}\hat{\mathbf{x}}_{F} \oplus {}^{F}\mathbf{e}$. To sense the state of a plane, our approach consists on obtaining two normal directions on the plane, to build from them the normal representing the surface orientation. A frame with its z axis fitting this normal direction and its origin being one of the sensed points on the surface, represents the location of the plane.

To obtain the state of an edge, we sense several points belonging to it. From them, we compute a frame which represent it. Such a frame has its x axis along the edge, and its origin at one of the sensed points.

In both cases, the trajectories are performed by using non-contact compliant and guarded motions using proximity sensors [9], [10]. These allow to sense features accomplishing the motion near unknown surfaces, while collisions with objects are avoided. Vertexes, are not directly measured, but computed from intersection of sensed edges. Details about the process to sense individual features with their location uncertainty can be found in [11] and about the control of compliant motions in [10].

4 Computing local constraints from geometric features

Before totally locating an object is necessary to recognize it. This can be made by matching sensed features in the scene to features in the model [12]. We advocate by a geometrical constraint-based recognition [7]. Exploiting local constraints, search space in the model and therefore computational burden is reduced. The constraints proposed and computed are: edge length, angle be-

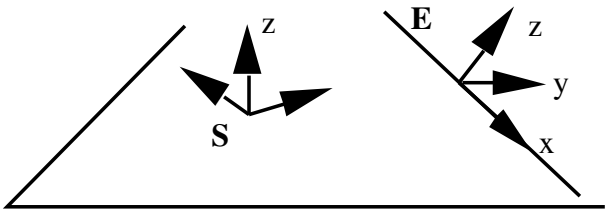

Figure 3: Representation of an edge on a plane

tween coplanar edges, angle between planar surfaces, distance between parallel edges and distance between parallel planes. Most of these constraints, as angles and distances between features, have the additional advantage of being robust to partial occlusions. As the sensed features have uncertainty, we explicitly consider it when we calculate the uncertainty of the constraints.

4.1 Edge length

We obtain the length as the distance between the two end points of the edge. Each end point is computed from the intersection between two coplanar edges which have been acquired with the sensors. Therefore, we first test the coplanarity between edges.

Coplanarity condition

We test coplanarity by checking that each edge belongs to the same plane. A right line is on a plane if the x axis of the frame attached to the line is on it (Figure 3). Taking into account the uncertainty of the sensed plane and edge, we can compute the relative transformation between their frames as:

$$^{S}\mathbf{x}_{E} = {}^{S}\hat{\mathbf{x}}_{E} \oplus {}^{E}\mathbf{e}^{S}$$

where:

$$\begin{array}{rcl} {}^{S}\hat{\mathbf{x}}_{E} & = & {}^{S}\hat{\mathbf{x}}_{W} \oplus {}^{W}\hat{\mathbf{x}}_{E} \\ {}^{E}\mathbf{e}^{S} & = & {}^{E}J_{S}{}^{S}\mathbf{e} + {}^{E}\mathbf{e} \end{array}$$

being ${}^{S}\mathbf{e}$ and ${}^{E}\mathbf{e}$ the location estimate errors of the surface and the edge, respectively. Let ${}^{S}\hat{\mathbf{x}}_{E} = (\hat{p}_x, \hat{p}_y, \hat{p}_z, \hat{\psi}, \hat{\theta}, \hat{\phi})^T$, $\mathbf{v}_1 = (0,0,1)^T$ the z unit vector of the plane frame estimate and $\mathbf{v}_2 = (\hat{n}_x, \hat{n}_y, \hat{n}_z)^T$ the x unit vector of the edge frame estimate expressed in the plane reference. Coplanarity conditions can be written with two equations:

$$\mathbf{v}_1^T \cdot \mathbf{v}_2 = 0 \tag{3}$$

$$\hat{p}_z = 0 \tag{4}$$

being $\cdot$ the dot product.

From (3) and expressing $\hat{n}_z$ as a function of ${}^{S}\hat{\mathbf{x}}_{E}$ (Appendix 1), we have the following condition:

$$\hat{n}_z = -sin\hat{\theta} = 0$$

Therefore, the coplanarity conditions are:

$$\begin{aligned} \hat{p}_z &= 0 \\ \hat{\theta} &= \pm n\pi \ ; \ n = 0, 1 \end{aligned} \tag{5}$$

Uncertainty of the condition represented by $Cov(p_z)$ and $Cov(\theta)$, are directly obtained from ${}^{E}\mathbf{e}^{S}$, because they respectively correspond to its third and fifth components. This information will be used in the recognition process.

Edge intersection

The intersection is obtained computing the point which simultaneously belongs to both x axes of the edge frames E1 and E2 (Figure 4). We calculate the intersection point refered to the edge the length of which we desire to measure. In the sequel development, it is E2.

Let ${}^{E1}\hat{\mathbf{x}}_{E2} = (\hat{p}_x, \hat{p}_y, \hat{p}_z, \hat{\psi}, \hat{\theta}, \hat{\phi})^T$ the relative transformation between $E1$ and $E2$ edges. Let $P1 = (\hat{p}_{x1}, 0, 0, 1)^T$ and $P2 = (\hat{p}_{x2}, 0, 0, 1)^T$the generic homogeneous coordinates of two points on the x axis of edges E1 and E2. The coordinates of a point on the x axis of E2 expressed in the frame E1 are ${}^{E1}H_{E2} \cdot (\hat{p}_{x2}, 0, 0, 1)^T$, being ${}^{E1}H_{E2}$ a homogeneous matrix as defined in Appendix 1. Equating both points, we obtain:

$$\begin{pmatrix} \hat{p}_{x1} \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \hat{n}_x\hat{p}_{x2} + \hat{p}_x \\ \hat{n}_y\hat{p}_{x2} + \hat{p}_y \\ \hat{n}_z\hat{p}_{x2} + \hat{p}_z \\ 1 \end{pmatrix} \tag{6}$$

The intersection point must simultaneously verify the equations (6), obtaining:

$$\hat{p}_{x2} = -\frac{\hat{p}_y}{\hat{n}_y} = -\frac{\hat{p}_z}{\hat{n}_z} \tag{7}$$

Note that if $E1$ and $E2$ are parallel, $\hat{n}_y = \hat{n}_z = 0$, and $\hat{p}_{x2}$ doesn't exist. If only one of them is zero, the x axis of $E2$ belongs to one of the coordinate planes of $E1(XY_{E1}, XZ_{E1})$ We always test the non null condition of (7). To directly obtain $\hat{p}_{x2}$ from ${}^{E1}\hat{\mathbf{x}}_{E2}$ (Appendix 1), we must compute:

$$\hat{p}_{x2} = -\frac{\hat{p}_y}{sin\hat{\phi}cos\hat{\theta}} = \frac{\hat{p}_z}{sin\hat{\theta}} \tag{8}$$

As P is a point, the frame attached to it can have any orientation. In particular we choose the same orientation as $E2$. Using the second condition of (8), the state estimate of P in $E2$ is:

$${}^{E2}\hat{\mathbf{x}}_P = \left(\ \frac{\hat{p}_z}{sin\hat{\theta}}, 0, 0, 0, 0, 0 \ \right)^T = f({}^{E1}\hat{\mathbf{x}}_{E2}) \tag{9}$$

To reach a full knowledge about the state of the point we need to obtain the estimate error ${}^{P}\mathbf{e}$ expressed in the P frame, so that:

$${}^{E2}\mathbf{x}_P = {}^{E2}\hat{\mathbf{x}}_P \oplus {}^{P}\mathbf{e}$$

which linearized is:

$$^{E2}\mathbf{x}_P \simeq {}^{E2}\hat{\mathbf{x}}_P + J_{2\oplus}\{{}^{E2}\hat{\mathbf{x}}_P, 0\} \cdot {}^{P}\mathbf{e} \tag{10}$$

being $J_{2\oplus}\{{}^{E2}\hat{\mathbf{x}}_P, 0\}$ the composition jacobian evaluated in $\{{}^{E2}\hat{\mathbf{x}}_P, 0\}$, as defined in [4]. Linearizing the equation (9) in ${}^{E1}\hat{\mathbf{x}}_{E2}$ we have:

$$^{E2}\mathbf{x}_P \simeq f({}^{E1}\hat{\mathbf{x}}_{E2}) + F^* \cdot ({}^{E1}\mathbf{x}_{E2} - {}^{E1}\hat{\mathbf{x}}_{E2}) \tag{11}$$

where:

$$F^* \triangleq \frac{\partial f}{\partial {}^{E1}\mathbf{x}_{E2}}\{{}^{E1}\hat{\mathbf{x}}_{E2}\} = \begin{pmatrix} 0 & 0 & \frac{1}{sin\hat{\theta}} & 0 & -\frac{\hat{p}_z cos\hat{\theta}}{sin^2\hat{\theta}} & 0 \\ & & O_{5\times 6} & & & \end{pmatrix}$$

Equating the linearized equations (10) and (11) we compute ${}^{P}\mathbf{e}$ as:

$${}^{P}\mathbf{e} = J_{2\oplus}^{-1}\{{}^{E2}\hat{\mathbf{x}}_P, 0\} \cdot F^* \cdot ({}^{E1}\mathbf{x}_{E2} - {}^{E1}\hat{\mathbf{x}}_{E2})$$

As ${}^{E1}\mathbf{x}_{E2} - {}^{E1}\hat{\mathbf{x}}_{E2} = J_{2\oplus}\{{}^{E1}\hat{\mathbf{x}}_{E2}, 0\} \cdot ({}^{E2}J_{E1}\,{}^{E1}\mathbf{e} + {}^{E2}\mathbf{e})$ (Appendix 2), last equation can be written as:

$${}^{P}\mathbf{e} = F \cdot ({}^{E2}J_{E1}\,{}^{E1}\mathbf{e} + {}^{E2}\mathbf{e}) \tag{12}$$

being $F \triangleq J_{2\oplus}^{-1}\{{}^{E2}\hat{\mathbf{x}}_P, 0\} \cdot F^* \cdot J_{2\oplus}\{{}^{E1}\hat{\mathbf{x}}_{E2}, 0\}$.

The covariance matrix $Cov({}^{P}\mathbf{e})$ is:

$$Cov({}^{P}\mathbf{e}) = F \cdot \left(\; {}^{E2}J_{E1} \cdot Cov({}^{E1}\mathbf{e}) \cdot {}^{E2}J_{E1}^T + Cov({}^{E2}\mathbf{e}) \; \right) \cdot F^T$$

This method can also be used to obtain a vertex location. In this case, we express the location ${}^{W}\mathbf{x}_P$ in world reference.

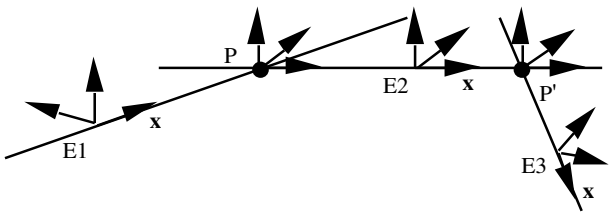

Figure 4: Frames attached to edges and intersection points to obtain the edge length

Edge length

The length of an edge is computed as the distance between two end points P and P' belonging to the edge (Figure 4). When the two frames attached to P and P' have been computed, we calculate the transformation ${}^{P}\hat{\mathbf{x}}_{P'} = (\hat{p}_x, \hat{p}_y, \hat{p}_z, \hat{\psi}, \hat{\theta}, \hat{\phi})^T$. The distance between both is:

$$l = \sqrt{\hat{p}_x^2 + \hat{p}_y^2 + \hat{p}_z^2} = f({}^{P}\hat{\mathbf{x}}_{P'}) \tag{13}$$

To estimate the length error we linearize (13) in ${}^{P}\hat{\mathbf{x}}_{P'}$:

$$\begin{aligned} l &\simeq f({}^{P}\hat{\mathbf{x}}_{P'}) + F^* \cdot ({}^{P}\mathbf{x}_{P'} - {}^{P}\hat{\mathbf{x}}_{P'}) = \\ &= f({}^{P}\hat{\mathbf{x}}_{P'}) + \varepsilon_l \end{aligned} \tag{14}$$

The error can be expressed as (Appendix 2):

$$\begin{aligned} \varepsilon_l &= F^* \cdot J_{2\oplus}\{{}^{P}\hat{\mathbf{x}}_{P'}, 0\} \cdot ({}^{P'}J_P \, {}^{P}\mathbf{e} + {}^{P'}\mathbf{e}) = \\ &= F_1 \cdot {}^{P}\mathbf{e} + F_2 \cdot {}^{P'}\mathbf{e} \end{aligned} \tag{15}$$

being

$$\begin{aligned} F^* &= \left(\tfrac{\hat{p}_x}{l}, \tfrac{\hat{p}_y}{l}, \tfrac{\hat{p}_z}{l}, 0, 0, 0 \right) \\ F_1 &\stackrel{\triangle}{=} F^* \cdot J_{2\oplus}\{{}^{P}\hat{\mathbf{x}}_{P'}, 0\} \cdot {}^{P'}J_P \\ F_2 &\stackrel{\triangle}{=} F^* \cdot J_{2\oplus}\{{}^{P}\hat{\mathbf{x}}_{P'}, 0\} \end{aligned}$$

The error covariance of the edge length calculated from the uncertainty of the constraints is:

$$Cov(\varepsilon_l) = F_1 \cdot Cov({}^{P}\mathbf{e}) \cdot F_1^T + F_2 \cdot Cov({}^{P'}\mathbf{e}) \cdot F_2^T$$

4.2 Angle between coplanar edges

To compute the angle between two edges we must test that both are coplanar (§4.1). If this condition is verified, the angle can be computed from the dot product of the x unit vector of each edge frame, both going away the intersection

point. Let $\mathbf{v}_1 = (1,0,0)^T$ the x unit vector in frame $E1$ and $\mathbf{v}_2 = (\hat{n}_x, \hat{n}_y, \hat{n}_z)^T$ the x unit vector of frame $E2$, expressed in frame $E1$. The dot product is:

$$\mathbf{v}_1^T \cdot \mathbf{v}_2 = \hat{n}_x = cos\alpha$$

being α the searched angle. From the relative transformation ${}^{E1}\hat{\mathbf{x}}_{E2}$ we obtain (Appendix 1):

$$\alpha = cos^{-1}(\hat{n}_x) = cos^{-1}(cos\hat{\phi}\, cos\hat{\theta}) = f({}^{E1}\hat{\mathbf{x}}_{E2}) \tag{16}$$

We choose the smallest angle which fits (16). From this solution it is impossible to know if the angle match with a concave or a convex corner. This ambiguity is solved with additional measures. For example, proximity sensors can inform in which side of the edge is the object. The angle α is a no-linear function of ${}^{E1}\mathbf{x}_{E2}$, and the error ε_α can be obtained in a similar way as in equations (15). That is, ε_α is computed as:

$$\varepsilon_\alpha = F_1 \cdot {}^{E1}\mathbf{e} + F_2 \cdot {}^{E2}\mathbf{e} \tag{17}$$

where:

$$\begin{array}{ll} F^* & = \left(\; 0,0,0,0, \frac{cos\hat{\phi}sin\hat{\theta}}{\Delta}, \frac{cos\hat{\theta}sin\hat{\phi}}{\Delta} \; \right) \\ F_1 & \stackrel{\triangle}{=} F^* \cdot J_{2\oplus}\{{}^{E1}\hat{\mathbf{x}}_{E2}, 0\} \cdot {}^{E2}J_{E1} \\ F_2 & \stackrel{\triangle}{=} F^* \cdot J_{2\oplus}\{{}^{E1}\hat{\mathbf{x}}_{E2}, 0\} \\ \Delta & \stackrel{\triangle}{=} \sqrt{1 - cos^2\hat{\phi}cos^2\hat{\theta}} \end{array}$$

The error covariance is:

$$Cov(\varepsilon_\alpha) = F_1 \cdot Cov({}^{E1}\mathbf{e}) \cdot F_1^T + F_2 \cdot Cov({}^{E2}\mathbf{e}) \cdot F_2^T \tag{18}$$

4.3 Angle between planes

The method to compute the angles between planes is similar to the method used in §4.2. The difference is that the searched angle is obtained from the z axis of frames attached to them (Figure 5).

At the same way that the angle between edges, the dot product between the z unit vectors of S1 frame, $\mathbf{v}_1 = (0,0,1)^T$, and of S2 frame expressed in S1, $\mathbf{v}_2 = (\hat{a}_x, \hat{a}_y, \hat{a}_z)^T$, yields the solution:

$$\mathbf{v}_1^T \cdot \mathbf{v}_2 = \hat{a}_z = cos\alpha$$

From the relative transformation ${}^{S1}\hat{\mathbf{x}}_{S2}$ between frames attached to the planes, we have (Appendix 1):

$$\alpha = cos^{-1}(cos\hat{\psi}cos\hat{\theta}) \tag{19}$$

If the planes are parallel, $\hat{a}_z = \pm 1$ and we have $\alpha = 0^o$ or $\alpha = 180^o$. The first solution corresponds to planes with the same orientation. The second one to

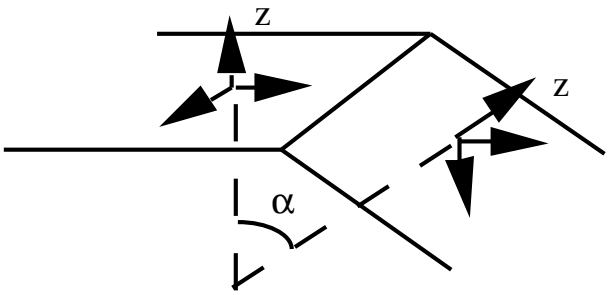

Figure 5: Angle between two planar surfaces

planes with opposite direction. The estimate error and the covariance have the same expressions that equations (17) and (18), changing ${}^{E1}\hat{\mathbf{x}}_{E2}$ by ${}^{S1}\hat{\mathbf{x}}_{S2}$, ${}^{E1}\mathbf{e}$ and ${}^{E2}\mathbf{e}$ by ${}^{S1}\mathbf{e}$ and ${}^{S2}\mathbf{e}$, and F^* by:

$$F^* = \left(\; 0, 0, 0, \frac{cos\hat{\theta}sin\hat{\psi}}{\Delta}, \frac{cos\hat{\psi}sin\hat{\theta}}{\Delta}, 0 \; \right)$$

being $\Delta \triangleq \sqrt{1 - cos^2\hat{\psi}cos^2\hat{\theta}}$.

4.4 Distance between parallel edges

To search the distance between parallel edges, belonging or not to the same surface, we first verify the parallelism condition. It can be written as:

$$\mathbf{v}_1^T \cdot \mathbf{v}_2 = \pm\, 1 \tag{20}$$

being $\mathbf{v}_1 = (1, 0, 0)^T$ and $\mathbf{v}_2 = (\hat{n}_x, \hat{n}_y, \hat{n}_z)^T$ the unit vectors along x axis of the $E1$ and $E2$ frames attached to the edges, both expressed in frame $E1$. From equation (20) we reach to:

$$\hat{n}_x = \pm\, 1 = cos\hat{\phi}cos\hat{\theta} = f({}^{E1}\hat{\mathbf{x}}_{E2}) \tag{21}$$

where $\hat{\phi}$ and $\hat{\theta}$ are obtained from the ${}^{E1}\hat{\mathbf{x}}_{E2} = (\hat{p}_x, \hat{p}_y, \hat{p}_z, \hat{\psi}, \hat{\theta}, \hat{\phi})^T$ transformation. The error in the parallelism condition $\varepsilon_{pc} \triangleq f({}^{E1}\mathbf{x}_{E2}) - f({}^{E1}\hat{\mathbf{x}}_{E2})$ can be computed by following similar steps that in the above paragraphs as:

$$\varepsilon_{pc} = F^* \cdot J_{2\oplus}\{{}^{E1}\hat{\mathbf{x}}_{E2}, 0\} \cdot ({}^{E2}J_{E1}\,{}^{E1}\mathbf{e} + {}^{E2}\mathbf{e})$$

being

$$F^* = (0, 0, 0, 0, -cos\hat{\phi}sin\hat{\theta}, -cos\hat{\theta}sin\hat{\phi})$$

The distance is calculated as the length of a normal segment to both edges.

$$d = \sqrt{\hat{p}_y^2 + \hat{p}_z^2}$$

The distance error is:

$$\varepsilon_d = F_1 \cdot {}^{E1}\mathbf{e} + F_2 \cdot {}^{E2}\mathbf{e}$$

where ${}^{E1}\mathbf{e}$ and ${}^{E2}\mathbf{e}$ are the errors of the edge locations estimate, and

$$\begin{aligned} F_1 &\stackrel{\triangle}{=} F^* \cdot J_{2\oplus}\{{}^{E1}\hat{\mathbf{x}}_{E2}, 0\} \cdot {}^{E2}J_{E1} \\ F_2 &\stackrel{\triangle}{=} F^* \cdot J_{2\oplus}\{{}^{E1}\hat{\mathbf{x}}_{E2}, 0\} \\ F^* &= \left(\ 0, \tfrac{\hat{p}_y}{d}, \tfrac{\hat{p}_z}{d}, 0, 0, 0 \ \right) \end{aligned}$$

The covariance matrix of distance error is computed as:

$$Cov(\varepsilon_d) = F_1 \cdot Cov({}^{E1}\mathbf{e}) \cdot F_1^T + F_2 \cdot Cov({}^{E2}\mathbf{e}) \cdot F_2^T$$

4.5 Distance between parallel planes

To calculate this constraint we must verify the parallelism condition of planes, given by the parallelism of their normals:

$$\mathbf{v}_1^T \cdot \mathbf{v}_2 = (0\ 0\ 1) \cdot \begin{pmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{pmatrix} = \hat{a}_z = \pm 1 \tag{22}$$

being $\mathbf{v}_1$ and $\mathbf{v}_2$ unit vectors along z axis directions. Condition (22), expressed as function of ${}^{S1}\hat{\mathbf{x}}_{S2}$ is (Appendix 1):

$$cos\hat{\psi}\ cos\hat{\theta} = \pm 1 \tag{23}$$

similar to condition (21).

The distance between two parallel planes is directly computed by means of the following expression:

$$d = |\hat{p}_z| = f({}^{S1}\hat{\mathbf{x}}_{S2}) \tag{24}$$

To estimate the error $\varepsilon_d = d - \hat{d}$, we proceed as in the previous paragraph, being the function derivative F^* of (24):

$$F^* = (0, 0, 1, 0, 0, 0)$$

5 Constraint-based recognition and localization

In a recognition and localization process, we use at first parametric information (shape descriptors) [4], [13], obtained from a 2D vision system . It has extensively been applied because of its easy use. However, it has some limitations, by example it is not robust to partial occlusions and may lead to ambiguity.

From the vision system, geometrical information is also obtained. To carry out a full recognition and localization, the acquisition of additional geometric

features by others sensors is required: either to compute geometrical constraints as explained in §4 or to eliminate ambiguity in localization.

As stated in the above paragraph, the constraints are computed with their uncertainty. In the recognition process we carry out a statistical distance test to compare the estimated value of a constraint with the constraints in the model. In the univariate case, if a stochastic variable x is characterized by a normal density function with mean $\hat{x}$ and variance σ^2, $N(\hat{x}, \sigma^2)$, the variable $D^2 = (x - \hat{x})^2/\sigma^2$ (D, Mahalanobis distance) is distributed as a χ^2 with 1 degree of freedom [14]. Therefore, knowing $\hat{x}$ and σ^2 of a computed constraint or condition (coplanarity, parallelism), we can know if it matchs with some in the model, allowing to accept or reject the object hypothesis.

We fix a reject percentage α, which in a χ^2 distribution, corresponds to a maximum value of Mahalanobis distance (D_M). For a estimated value of a constraint $\hat{c}$, with variance $Cov(c)$ computed as shown in §4, we accept a value in the model c_m, if it verifies:

$$|c_m - \hat{c}| \leq \sqrt{Cov(c)} \cdot D_M \tag{25}$$

The process is repeated for all the values in the model.

6 Conclusion

To observe geometric features, active sensing strategies using proximity sensors carried by the robot are well suited. When using active sensing, we try to exploit the discriminant power of some features, which may be occluded to other sensors.

We have proposed a probabilistic model to represent geometric features. The estimated location of a feature is the estimated location of a reference system attached to it, and its uncertainty is represented by a differential transformation, characterized by its mean and its covariance matrix.

As proximity sensors are used in active sensing, the measurement device and the robot location errors have been explicitly considered in the model. Previous information is obtained by a 2D vision system and the sensorial system takes advantage of it to drive the proximity sensors towards features to be observed. Some methods to acquire geometric features with proximity sensors have been outlined.

We have computed constraints from sensed geometric features. The uncertainty of the constraints has been obtained taking into account the feature uncertainty. In the recognition process, matching by means of a statistical distance test between acquired and modeled features and constraints is achieved.

This work will be extended to acquire other features (circles, non-planar surfaces,...) and to compute constraints from them. Active sensing with camera-in-hand is also being studied.

Acknowledgements

This work was partially supported by project PA86-0028 of the Comisión Interministerial de Ciencia y Tecnología (CICYT) and by project IT-5/90 of the Diputación General de Aragón (CONAI).

References

[1] R. A. Brooks. Visual map making for a mobile robot. In *IEEE Int. Conf. on Robotics Research*, pages 824–829, 1985.

[2] H.F Durrant-Whyte. *Integration Coordination and Control of Multi-Sensor Robot Systems*. Kluwer Academic Pub., Massachusetts, 1988.

[3] 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.

[4] 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, December 1991.

[5] 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.

[6] 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.

[7] W.E.L. Grimson. The combinatorics of local constraints in model-based recognition and localization from sparse data. *Journal of the Association for Computing Machinery*, 33(4):658–686, 1986.

[8] R.P. Paul. *Robot Manipulators: Mathematics, Programming, and Control*. MIT Press, Cambridge, Mass., 1981.

[9] C. Sagüés, L. Montano, and J. Neira. Guarded and compliant motions using force and proximity sensors. In *Int. Workshop on Sensorial Integration for Industrial Robots*, pages 274–280, Zaragoza-Spain, 1989.

[10] L. Montano and C. Sagüés. Non-contact compliant robot motions: Dynamic behaviour and application to feature localization. In *IMACS-MCTS Symposium*, Lille, France, May 1991.

[11] C. Sagüés and L. Montano. Active sensing strategies with non-contact compliant motions for constarint-based recognition. In *IFAC/IFIP/IMACS-Symposium on Robot Control SYROCO'91*, Vienna, Austria, September 1991 (To appear).

[12] O. D. Faugeras and M. Hebert. The representation recognition, and locating of 3D objects. *Int. J. Robotics Research*, 5(3):27–52, 1986.

[13] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[14] S. Ríos. *Métodos Estadísticos*. Ediciones del Castillo S.A., Madrid, Spain, 1977.

APPENDIX 1. Homogeneous matrices and location vectors

A homogeneous matrix H, representing a transformation between reference systems [8], is expressed as:

$$H = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} R & \mathbf{p} \\ 0 & 1 \end{pmatrix}$$

where R is a orthogonal rotation matrix and $\mathbf{p}$ is a position vector. A location vector is expressed by:

$$\mathbf{x} = (p_x, p_y, p_z, \psi, \theta, \phi)^T$$

where (p_x, p_y, p_z) represent the position and (ψ, θ, ϕ) are the orientation Yaw-Pitch-Roll angles:

$$\psi \in (-\pi, \pi], \;\; \theta \in (-\frac{\pi}{2}, \frac{\pi}{2}], \;\; \phi \in (-\pi, \pi]$$

From a location vector, the elements of the homogeneous matrix used in the paper are:

$$\begin{aligned} n_x = cos\phi cos\theta, \quad n_y = sin\phi cos\theta \\ n_z = -sin\theta, \quad a_z = cos\psi cos\theta \end{aligned}$$

APPENDIX 2. Relating errors between frames

Given the transformations ${}^{F1}\mathbf{x}_W$ and ${}^{W}\mathbf{x}_{F2}$:

$${}^{F1}\mathbf{x}_W = {}^{F1}\mathbf{e} \oplus {}^{F1}\hat{\mathbf{x}}_W, \;\; {}^{W}\mathbf{x}_{F2} = {}^{W}\hat{\mathbf{x}}_{F2} \oplus {}^{F2}\mathbf{e}$$

we deduce the relative transformation ${}^{F1}\mathbf{x}_{F2}$ and its estimate error, expressed in one of the frames $(F2)$:

$$\begin{aligned} {}^{F1}\mathbf{x}_{F2} &= {}^{F1}\mathbf{e} \oplus {}^{F1}\hat{\mathbf{x}}_W \oplus {}^{W}\hat{\mathbf{x}}_{F2} \oplus {}^{F2}\mathbf{e} = \\ &= ({}^{F1}\hat{\mathbf{x}}_W \oplus {}^{W}\hat{\mathbf{x}}_{F2}) \oplus ({}^{F2}J_{F1}\,{}^{F1}\mathbf{e} \oplus {}^{F2}\mathbf{e}) = \\ &= {}^{F1}\hat{\mathbf{x}}_{F2} \oplus {}^{F2}\mathbf{e}^{F1} \end{aligned} \tag{26}$$

As ${}^{F2}\mathbf{e}$ and ${}^{F1}\mathbf{e}$ are incremental transformations, ${}^{F2}\mathbf{e}^{F1} \simeq {}^{F2}J_{F1}\,{}^{F1}\mathbf{e} + {}^{F2}\mathbf{e}$.

Sometimes, we have the transformation between frames expressed as:

$$ {}^{F1}\mathbf{x}_{F2} \simeq {}^{F1}\hat{\mathbf{x}}_{F2} + {}^{F2}\varepsilon^{F1} \tag{27} $$

Taking into account that ${}^{F2}\hat{\mathbf{e}}^{F1} = 0$, the relation between ${}^{F2}\mathbf{e}^{F1}$ and ${}^{F2}\varepsilon^{F1}$ is obtained linearizing equation (26) in $\{{}^{F1}\hat{\mathbf{x}}_{F2}, {}^{F2}\hat{\mathbf{e}}^{F1}\}$:

$$ \begin{aligned} {}^{F1}\mathbf{x}_{F2} &\simeq {}^{F1}\hat{\mathbf{x}}_{F2} + J_{1\oplus}\{{}^{F1}\hat{\mathbf{x}}_{F2}, {}^{F2}\hat{\mathbf{e}}^{F1}\} \cdot ({}^{F1}\hat{\mathbf{x}}_{F2} - {}^{F1}\hat{\mathbf{x}}_{F2}) \\ &\quad + J_{2\oplus}\{{}^{F1}\hat{\mathbf{x}}_{F2}, {}^{F2}\hat{\mathbf{e}}^{F1}\} \cdot ({}^{F2}\mathbf{e}^{F1} - {}^{F2}\hat{\mathbf{e}}^{F1}) = \\ &= {}^{F1}\hat{\mathbf{x}}_{F2} + J_{2\oplus}\{{}^{F1}\hat{\mathbf{x}}_{F2}, 0\} \cdot {}^{F2}\mathbf{e}^{F1} \end{aligned} \tag{28} $$

being $J_{1\oplus}\{\hat{\mathbf{x}}, \hat{\mathbf{e}}\}$ and $J_{2\oplus}\{\hat{\mathbf{x}}, \hat{\mathbf{e}}\}$ the composition jacobians as defined in [4].

Equating (27) and (28) we reach to:

$$ \begin{aligned} {}^{F2}\varepsilon^{F1} &= {}^{F1}\mathbf{x}_{F2} - {}^{F1}\hat{\mathbf{x}}_{F2} = \\ &= J_{2\oplus}\{{}^{F1}\hat{\mathbf{x}}_{F2}, 0\} \cdot ({}^{F2}J_{F1}\,{}^{F1}\mathbf{e} + {}^{F2}\mathbf{e}) \end{aligned} $$