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

Internal Report: 1992-01

Active sensing strategies with non-contact compliant motions for contraint based recognition

Sagüés C., Montano L.

If you want to cite this report, please use the following reference instead:
Active sensing strategies with non-contact compliant motions for contraint based recognition, Sagüés C., Montano L., *Robot Control, Pergamon Press*, ISBN 0 08 041276 9, pages 475-480, 1992.

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 STRATEGIES WITH NON-CONTACT COMPLIANT MOTIONS FOR CONSTRAINT BASED RECOGNITION

SAGÜÉS Carlos, MONTANO Luis

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

Abstract. Data driven active sensing strategies for the adquisition of geometric features are presented. In a object recognition and localization process, these techniques allow to exploit the discriminant power of the features although they are partially occluded to some fixed sensors. Geometric features are represented taking into account the location uncertainty due to the measurement errors. To drive the sensors to observe the features, a control mechanism is required. We use non-contact compliant motions with proximity sensors to sense features and reduce uncertainty when their location is partially known. The control algorithms have been implemented in a experimental multisensorial system, using a PUMA 560 robot.

Keywords. Robots, sensors, computer control, compliant motions, active sensing.

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 multisensor capabilities. We have designed and implemented a multisensorial robotic system named APRIL (Montano, 1987), using a PUMA 560 robot, a 2D vision system, a force/torque sensor-in-wrist and six infrared proximity sensors on the end-effector. Two microcameras and two proximity laser sensors are being mounted on the robot hand.

In this paper we propose active sensing techniques (Ellis, 1987; Grimson, 1990; Lee, 1990) using proximity sensors on the robot gripper, to acquire and estimate feature location (vertexes, edges and planar surfaces) of polyhedral objects. The main advantage of these techniques appears because the number of observations needed to identify and locate objects can be optimized; in this sense they allow to exploit the discriminant power of some features, although they are partially occluded.

To carry out these techniques we have implemented guarded and compliant motions. The motion control scheme is based on the idea of generalized damping (Mason, 1982), extending its application to *non-contact* motions (Montano, 1991a).

Before undertaking active sensing strategies we present the compliant motion control scheme and we outline some conclusions about its dynamic behaviour. In the following paragraphs the representation of the geometric features to be observed, the proximity sensor model and the technique to estimate the location of the features are presented.

NON-CONTACT COMPLIANT MOTION CONTROL

Non-contact compliant motions based on proximity sensor information, have been implemented in the APRIL robot programming system. We have mounted six infrared sensors (three on each finger), located as follows: two on the front of the fingers (RF, LF), two on the internal face (RC, LC) and two on the external face (RE, LE) as shown in Fig. 1.

Motion control is achieved by correcting nominal velocities programmed on the PUMA robot, as a function of the information obtained from proximity sensors (Montano, 1991a). Each sampling period the PUMA controller adds (integrates) the velocity correction computed by our controller. In Fig. 2 the damping control scheme for non-contact compliance is shown.

Mathematically it can be expressed in the following way:

$$\boldsymbol{\delta}\dot{\boldsymbol{X}}_c = \mathbf{C}\cdot(\mathbf{d}_s - \mathbf{d}_c)\ ;\ \ \dot{\boldsymbol{X}}_c = \dot{\boldsymbol{X}}_s + \boldsymbol{\delta}\dot{\boldsymbol{X}}_c \qquad (1)$$

where each term is defined as follows:

$\boldsymbol{\delta}\ \dot{\boldsymbol{X}}_c = (\delta\dot{x}, \delta\dot{y}, \delta\dot{z}, \delta\dot{\theta}_x, \delta\dot{\theta}_y, \delta\dot{\theta}_z)^T$: correction velocity in the compliance frame
$\mathbf{C}$: correction matrix (control parameter)

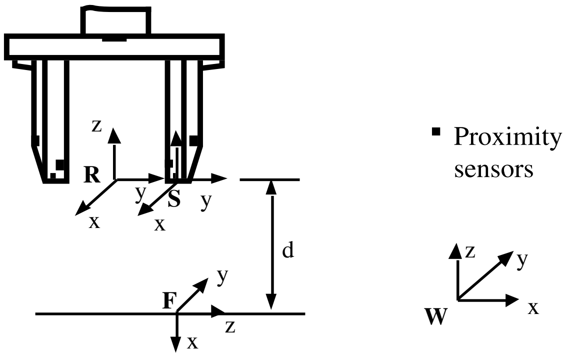


Figure 1: Proximity sensors on the robot gripper and frames attached

$\mathbf{d}_s = (d^s_{RF}, d^s_{LF}, d^s_{RC}, d^s_{LC}, d^s_{RE}, d^s_{LE})T$: distance setpoint
$\mathbf{d}_c = (d_{RF}, d_{LF}, d_{RC}, d_{LC}, d_{RE}, d_{LE})^T$: measured distance in the compliance frame
$\dot{\boldsymbol{X}}_c = (\dot{x}, \dot{y}, \dot{z}, \dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z)^T$: corrected velocity
$\dot{\boldsymbol{X}}_s = (\dot{x}^s, \dot{y}^s, \dot{z}^s, \dot{\theta}^s_x, \dot{\theta}^s_y, \dot{\theta}^s_z)^T$: velocity setpoint.

Selection of the values of the elements of **C** matrix depends on the task and on the proximity sensors layout. That is, we associate the velocity correction in the compliance frame with a specific set of sensors, suitable for the completion of the task. Additionally, the **C** matrix influences on the dynamic behaviour and the system stability. Below, this influence is exposed.

Open loop system model

With the aim of making an analytical study of the transient response and the stability of the control system as a function of the parameters of the **C** matrix, we have obtained its mathematical model by identification techniques. The open loop plant identified is remarked in Fig. 2. It includes the robot controller (motion generation, coordinate transformation and joint servoing) and the proximity sensors. We restrict our study to tasks in which only frontal sensors are used, therefore, velocity corrections along z and around x axes of robot tool frame R (Fig. 1) are involved.

Robots are non-linear, coupled and with time-varying parameter systems. However, non-contact motions in the refered tasks are normally accomplished at low speeds. This has allowed us to use a time-invariant linear model as an approximation to the real system. Anyway, we considered this variation by obtaining models for two extreme locations (far, near) in the robot workspace.

By using identification methods based in the least-squares criterion we obtained the model which expressed as state space equations is:

$$\begin{array}{lcl} \mathbf{x}(k+1) & = & \mathbf{F}\cdot\mathbf{x}(k) + \mathbf{G}\cdot\mathbf{u}(k) \\ \mathbf{y}(k) & = & \mathbf{H}\cdot\mathbf{x}(k) \end{array} \tag{2}$$

where $\mathbf{x}(k)$ is the state vector, $\mathbf{y}(k)$ is the output vector which is the measured distance $(d_{RF}, d_{LF})^T$, and $\mathbf{u}(k)$ is the input vector which is the velocity corrections $(\delta\dot{z}, \delta\dot{\theta}_x)^T$. The **F**, **G** and **H** matrices, that can be expressed with decoupled translational and rotational parts, are:

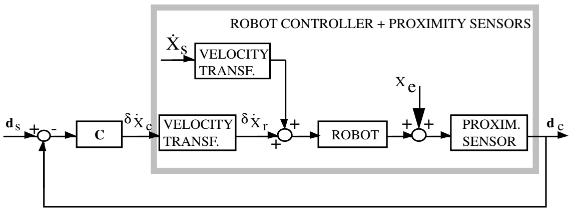


Figure 2: Damping control for non-contact compliance

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_\theta \end{pmatrix}; \quad \mathbf{G} = \begin{pmatrix} \mathbf{G}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_\theta \end{pmatrix}$$
$$\mathbf{H} = (\mathbf{H}_z \;\; \mathbf{H}_\theta)$$

where the expressions of $\mathbf{F}_\alpha$ and $\mathbf{G}_\alpha$ $(\alpha = z, \theta)$ can, in our model, be extended as:

$$\mathbf{F}_\alpha = \begin{pmatrix} -a_1 & -a_2 & -a_3 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$
$$\mathbf{G}_\alpha = (1\;0\;0\;0\;0\;0)^T \tag{3}$$
$$\mathbf{H}_z = \begin{pmatrix} 0 & 0 & 0 & b_1 & b_2 & 0 \\ 0 & 0 & 0 & b_1 & b_2 & 0 \end{pmatrix}$$
$$\mathbf{H}_\theta = \begin{pmatrix} 0 & 0 & 0 & b_1 & b_2 & b_3 \\ 0 & 0 & 0 & -b_1 & -b_2 & -b_3 \end{pmatrix}$$

The a_i and b_i parameters are given in Table 1, for near and far robot locations.

Closed loop dynamic behaviour

From the open loop system model obtained, we have studied the system stability as a function of the con-

Table 1: System parameters estimate

		M_{11}	M_{11}	M_{12}	M_{12}
		far	near	far	near
a_1	P	-1.0334	-1.9676	-1.6817	-1.0544
	sd	0.0474	0.0572	0.0587	0.0928
a_2	P	-0.0001	1.2004	0.6250	0.1980
	sd	0.0707	0.0831	0.0954	0.1011
a_3	P	0.0343	-0.2326	0.0573	-0.1433
	sd	0.0303	0.0287	0.0385	-0.026
b_1	P	0.1583	0.1801	0.0683	-0.0062
	sd	0.0308	0.0561	0.0192	0.0664
b_2	P	0.7787	0.0809	-0.2889	-0.0994
	sd	0.0377	0.0763	0.0306	-0.0889
b_3	P	0	0	0.0389	-0.6237
	sd	0	0	0.0291	0.0862

trol parameters, which are the elements of the **C** matrix. The 2x2 **C** matrix is:

$$\mathbf{C} = \begin{pmatrix} c_1 & c_1 \\ -c_2 & c_2 \end{pmatrix}$$

If we let $\mathbf{y}_s(k)$ and $\mathbf{y}(k)$ the setpoint and the measured distances, the velocity correction $\mathbf{u}(k)$ (input to the system) is computed as:

$$\mathbf{u}(k) = \mathbf{C}(\mathbf{y}_s(k) - \mathbf{y}(k)) \tag{4}$$

Substituting (4) into (2), we reach to:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{C}(\mathbf{y}_s(k) - \mathbf{y}(k)) = \\ &= (\mathbf{F} - \mathbf{G}\mathbf{C}\mathbf{H})\mathbf{x}(k) + \mathbf{G}\mathbf{C}\mathbf{y}_s(k) \end{aligned} \tag{5}$$

System behaviour depends on the eigenvalues of $\mathbf{F}^* \stackrel{\triangle}{=} \mathbf{F} - \mathbf{G}\mathbf{C}\mathbf{H}$:

$$\mathbf{F}^* = \begin{pmatrix} \mathbf{F}^*_z & \mathbf{0} \\ \mathbf{0} & \mathbf{F}^*_\theta \end{pmatrix} \tag{6}$$

Eigenvalues of the traslational part ($\mathbf{F}^*_z$) depend on c_1 and those of the rotational part ($\mathbf{F}^*_\theta$) depend on c_2. From the root locus analysis we have deduced the limit values estimate of the c_i parameters obtained for the two extreme locations (far and near) which render the system stable:

$$\mathbf{C}_{far} = \begin{pmatrix} 0.178 & 0.178 \\ -0.196 & 0.196 \end{pmatrix}$$

$$\mathbf{C}_{near} = \begin{pmatrix} 0.164 & 0.164 \\ -0.224 & 0.224 \end{pmatrix}$$

From similar analysis we have also obtained the c_i values which give no overshooting ($c_1 <= 0.031$, $c_2 <= 0.05$) when approaching to an object.

The range perception system and the compliant motions presented are used in the following paragraphs to extract geometric features in the recognition and localization process.

REPRESENTING 3-D OBJECT GEOMETRIC FEATURES

We deal with polyhedral objects and we consider them to be built by vertexes, edges and planar surfaces. We use a frame (F) attached to each feature (Tardós, 1991), expressing the generic location of F frame 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 axes of the frame. 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 Tardós (1991) a technique based on selection matrices to take the parameters just needed is proposed.

To represent the location uncertainty we use an incremental transformation

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

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 ${}^W\hat{\mathbf{x}}_F \stackrel{\triangle}{=} E\{{}^W\mathbf{x}_F\} = (\hat{p}_x, \hat{p}_y, \hat{p}_z, \hat{\psi}, \hat{\theta}, \hat{\phi})^T$ 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 (Paul, 1981), and $\oplus$ is the operator to compose transformations when they are represented as location vectors (Smith, 1988). Assuming the hypothesis of gaussian white noise, the location uncertainty is 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 $Cov({}^F\mathbf{e})$.

PROXIMITY SENSOR MODEL

To model the sensors in a multisensorial system, questions such as the system state, the sensor state, the sensor utility, the acquisition cost, the functioning requirements and the relations with others sensors should be considered. In this work we deal with a measurement model of the proximity sensors that takes into account measurement device and robot positioning errors. The model that we search, relates the distance measurement from the sensor to a point (represented by the location vector of the F frame attached to the point in the sensor frame ${}^S\mathbf{x}_F$) with the location of the feature in the world frame ${}^W\mathbf{x}_F$ (Fig. 1).

Let ${}^S\hat{\mathbf{x}}_F$ the location estimate of F in the sensor frame and ${}^W\hat{\mathbf{x}}_R$ location estimate of R in the world frame. Their true values will be

$$\begin{aligned} {}^S\mathbf{x}_F &= {}^S\hat{\mathbf{x}}_F \oplus {}^F J_S {}^S\mathbf{e}_d \\ {}^W\mathbf{x}_R &= {}^W\hat{\mathbf{x}}_R \oplus {}^R\mathbf{e}_r \end{aligned}$$

being ${}^F J_S$ the transformation jacobian, and ${}^S\mathbf{e}_d$ and ${}^R\mathbf{e}_r$ the incremental transformations representing the measurement device error in the sensor frame

S and the robot position and orientation error in the robot frame R, respectively.

The location of F in world reference is:

$$\begin{aligned} {}^W\mathbf{x}_F &= ({}^W\hat{\mathbf{x}}_R \oplus {}^R\mathbf{e}_r) \oplus ({}^R\mathbf{x}_S \oplus {}^S\hat{\mathbf{x}}_F \oplus {}^F J_S {}^S\mathbf{e}_d) \\ &= ({}^W\hat{\mathbf{x}}_R \oplus {}^R\hat{\mathbf{x}}_F) \oplus ({}^F J_R {}^R\mathbf{e}_r + {}^F J_S {}^S\mathbf{e}_d) \\ &= {}^W\hat{\mathbf{x}}_F \oplus {}^F\mathbf{e}_s \qquad (7) \end{aligned}$$

where ${}^F\mathbf{e}_s$ is the global sensor error due to the measurement device and robot that carries it and ${}^F J_R$ and ${}^F J_S$ are the transformation Jacobians. Below these expressions are used to find the feature state (location) from the sensor measurements.

To obtain the total characterization of the measurement error, we compute the error covariance matrix as:

$$\begin{aligned} &Cov({}^F\mathbf{e}_s) = Cov({}^F J_R\ {}^R\mathbf{e}_r + {}^F J_S\ {}^S\mathbf{e}_d) = \\ &= {}^F J_R \cdot Cov({}^R\mathbf{e}_r) \cdot {}^F J_R^T + {}^F J_S \cdot Cov({}^S\mathbf{e}_d) \cdot {}^F J_S^T \end{aligned}$$

because ${}^R\mathbf{e}_r$ and ${}^S\mathbf{e}_d$ are independent. $Cov({}^R\mathbf{e}_r)$ and $Cov({}^S\mathbf{e}_d)$ are supposed to be known; in this sense, we are characterizing both errors.

ESTIMATING LOCATION OF FEATURES

We assume that to sense geometric features with proximity sensors, previous information from other sensors or location hypothesis generated in the recognition process are given. Some features not observable by other fixed sensors can be explored by proximity sensors carried by the robot end-effector. We propose data driven strategies for features acquisition performed by using non-contact guarded and compliant motions which allow to sense features accomplishing the motion near unknown surfaces, while collisions with objects is avoided.

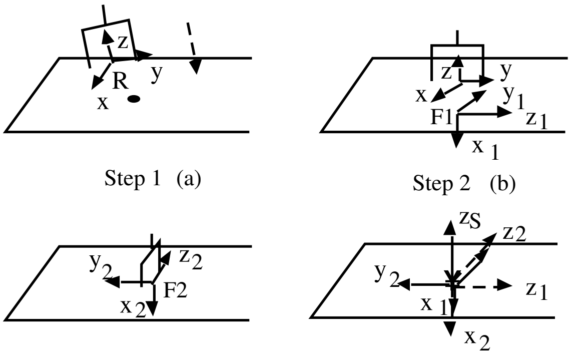
The sensed features are edges and planar surfaces. In both, the state to be acquired is the location in the world reference of a frame attached to it, named ${}^W\mathbf{x}_F$. Vertexes, are not directly measured, but computed from intersection of sensed edges.

To sense the state of a plane, our approach consists on obtaining two normal directions on the plane, to build from them the normal to the plane, representing the surface orientation. A frame with its z axis fitting this normal direction and its origin being one of the sensed points represents the feature state. To obtain the state of an edge, we sense two points belonging to it. From them, we compute a frame with its x axis along the edge, and its origin at one of the sensed points.

We describe now the strategies to acquire a plane and an edge.

Locating a plane

To find a frame attached to a surface we proceed as follows (Fig. 3):



Step 1 (a)

Step 2 (b)

Step 3 (c)

Step 4 (d)

Figure 3: Sequence to sense a plane

1. Approach towards a point of the hypothesized surface location using a compliant motion with a given distance setpoint. This technique makes possible to place both sensors at the same distance to the plane (Fig. 3a) when the height and shape of the object are unknown.

2. Store the frame location (${}^W\mathbf{x}_{F1}$) attached to a parallel direction to the sensed surface. We impose that this frame have the z axis along the direction joining the measured points by the two sensors and its origin in the middle of both points (Fig. 3b).

3. Rotate the end-effector by 90 degrees around z axis of the tool reference, using a compliant motion. As in step 2, store the frame location (${}^W\mathbf{x}_{F2}$) attached to a parallel direction to the sensed surface (Fig. 3c).

4. Compute the frame location ${}^W\mathbf{x}_S$ attached to the planar surface from ${}^W\mathbf{x}_{F1}$ and ${}^W\mathbf{x}_{F2}$ (Fig. 3d).

Computing ${}^W\mathbf{x}_S$. We obtain ${}^W\mathbf{x}_S$ by rotating F2 frame, 90 degrees around the z axis of F1 frame. Locations ${}^W\mathbf{x}_{F1}$ and ${}^W\mathbf{x}_{F2}$ are computed by using (7) equation.

To calculate ${}^W\mathbf{x}_S$ we need to know the transformation between F1 and F2, which is obtained from equation (12) in Appendix. Let R a 90 degrees rotation transformation around z axis. ${}^W\hat{\mathbf{x}}_S$ is obtained by applying R to ${}^{F1}\hat{\mathbf{x}}_{F2}$ expressed in frame $F1$:

$$ {}^W\hat{\mathbf{x}}_S = {}^W\hat{\mathbf{x}}_{F1} \oplus (R \oplus {}^{F1}\hat{\mathbf{x}}_{F2}) $$

The real value of ${}^W\mathbf{x}_S$ will be:

$$ {}^W\mathbf{x}_S = {}^W\hat{\mathbf{x}}_{F1} \oplus (R \oplus {}^{F1}\hat{\mathbf{x}}_{F2} \oplus {}^{F2}\mathbf{e}^{F1}) $$

where ${}^{F2}\mathbf{e}^{F1}$ is the global error between the $F1$ and $F2$ frames as defined in (12) of Appendix.

Letting ${}^W\hat{\mathbf{x}}_S \triangleq {}^W\hat{\mathbf{x}}_{F1} \oplus R \oplus {}^{F1}\hat{\mathbf{x}}_{F2}$ and ${}^S\mathbf{e} \triangleq {}^{F2}\mathbf{e}^{F1}$, we reach to:

$$ {}^W\mathbf{x}_S \quad = {}^W\hat{\mathbf{x}}_S \oplus {}^S\mathbf{e} $$

The covariance matrix of ${}^S\mathbf{e}$ can be obtained as:

$$ Cov({}^S\mathbf{e}) = {}^{F2}J_{F1} \cdot Cov({}^{F1}\mathbf{e}) \cdot {}^{F2}J_{F1}^T + Cov({}^{F2}\mathbf{e}) $$

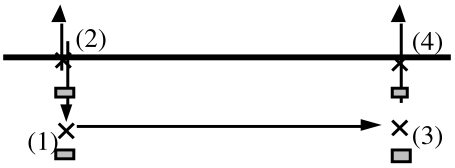


Figure 4: Sensing an edge

which characterize the plane location uncertainty.

Locating an edge

To locate an edge we sense two points on it. We choose them as close as possible to the known ends, because the larger the distance between points the smaller the location uncertainty of the edge. The proposed strategy to find an edge proceeds in the following steps (Fig. 4):

1. Approach towards a point of the hypothesized surface location using a compliant motion with a given distance setpoint. Locate the robot end-effector normal to the plane containing the edge. To perform it, we use the same motions than described to detect a plane.

2. By using a compliant motion, approach to a point located near to one end of the edge (point 1).

3. Perform a guarded motion, through an approximately normal direction to the edge, monitoring with one sensor the measured distance to the plane. A point is stored (point 2)when a sudden distance change is detected.

4. Come back to the initial point 1.

5. Start a compliant motion towards the other end of the edge (point 3), and by means of a similar motion than 4, locate the second point (4).

Detection of the two points suffice to obtain the location of the frame attached to the edge (${}^W\mathbf{x}_E$).

Computing ${}^W\mathbf{x}_E$. Let ${}^W\mathbf{x}_{P1}$ and ${}^W\mathbf{x}_{P2}$ the frame locations attached to the points P1 and P2 obtained as expressed in (7). We attach to the edge a frame (E) whose origin is, for example, P1. From P1 position $(\hat{p}_{x1}, \hat{p}_{y1}, \hat{p}_{z1})$ and P2 position $(\hat{p}_{x2}, \hat{p}_{y2}, \hat{p}_{z2})$ we compute its orientation, in such a way that its x axis is along the edge. This location ${}^W\hat{\mathbf{x}}_E = f({}^W\hat{\mathbf{x}}_{P1}, {}^W\hat{\mathbf{x}}_{P2})$ is a non-linear function of ${}^W\mathbf{x}_{P1}$ and ${}^W\mathbf{x}_{P2}$ and it can be computed as:

$$ {}^W\hat{\mathbf{x}}_E = (\hat{p}_{x1},\ \hat{p}_{y1},\ \hat{p}_{z1},\ 0,\ \alpha,\ \beta)^T \tag{8}$$

being $\alpha = atan2\ \frac{\hat{p}_{z1}-\hat{p}_{z2}}{\sqrt{(\hat{p}_{x2}-\hat{p}_{x1})^2+(\hat{p}_{y2}-\hat{p}_{y1})^2}}$ and $\beta = atan2\ \frac{\hat{p}_{y2}-\hat{p}_{y1}}{\hat{p}_{x2}-\hat{p}_{x1}}$.

To get a full knowledge about ${}^W\mathbf{x}_E$, we need to obtain the estimate error ${}^E\mathbf{e}$, so that:

$$ {}^W\mathbf{x}_E = {}^W\hat{\mathbf{x}}_E \oplus {}^E\mathbf{e} \tag{9}$$

To compute ${}^E\mathbf{e}$, that is a function of the known position errors ${}^{P1}\mathbf{e}$ and ${}^{P2}\mathbf{e}$ of the frames attached to the sensed points, we linearize ${}^W\mathbf{x}_E = f({}^W\mathbf{x}_{P1}, {}^W\mathbf{x}_{P2})$ in $\{{}^W\hat{\mathbf{x}}_{P1}, {}^W\hat{\mathbf{x}}_{P2}\}$, and using the results of Appendix, we obtain:

$$\begin{aligned} {}^W\mathbf{x}_E &\simeq {}^W\hat{\mathbf{x}}_E + F_1^*({}^W\mathbf{x}_{P1} - {}^W\hat{\mathbf{x}}_{P1}) + \\ &\quad F_2^*({}^W\mathbf{x}_{P2} - {}^W\hat{\mathbf{x}}_{P2}) \\ &= {}^W\hat{\mathbf{x}}_E + F_1^* \cdot J_{2\oplus}\{{}^W\hat{\mathbf{x}}_{P1}, 0\} \cdot {}^{P1}\mathbf{e} + \\ &\quad F_2^* \cdot J_{2\oplus}\{{}^W\hat{\mathbf{x}}_{P2}, 0\} \cdot {}^{P2}\mathbf{e} \end{aligned} \tag{10}$$

where

$$ F_1^* \triangleq \frac{\partial f}{\partial {}^W\mathbf{x}_{P1}}\{{}^W\hat{\mathbf{x}}_{P1}, {}^W\hat{\mathbf{x}}_{P2}\}$$

$$ F_2^* \triangleq \frac{\partial f}{\partial {}^W\mathbf{x}_{P2}}\{{}^W\hat{\mathbf{x}}_{P1}, {}^W\hat{\mathbf{x}}_{P2}\}$$

whose expressions are:

$$ F_1^* = \begin{pmatrix} & I_{3\times3} & & O_{3\times3} \\ 0 & 0 & 0 & \\ f_{51} & f_{52} & f_{53} & O_{3\times3} \\ f_{61} & f_{62} & 0 & \end{pmatrix}$$

$$ F_2^* = \begin{pmatrix} & O_{3\times3} & & O_{3\times3} \\ 0 & 0 & 0 & \\ -f_{51} & -f_{52} & -f_{53} & O_{3\times3} \\ -f_{61} & -f_{62} & 0 & \end{pmatrix}$$

where $I_{3\times3}$ is the unit matrix, $O_{3\times3}$ is the null matrix and

$$\begin{aligned} f_{51} &\triangleq \frac{(\hat{p}_{x1}-\hat{p}_{x2})(\hat{p}_{z2}-\hat{p}_{z1})}{\sqrt{D_1}\cdot D_2} \\ f_{52} &\triangleq \frac{(\hat{p}_{y2}-\hat{p}_{y1})(\hat{p}_{z1}-\hat{p}_{z2})}{\sqrt{D_1}\cdot D_2} \\ f_{53} &\triangleq \frac{\sqrt{D_1}}{D_2} \\ f_{61} &\triangleq \frac{\hat{p}_{y2}-\hat{p}_{y1}}{D_1} \\ f_{62} &\triangleq \frac{\hat{p}_{x1}-\hat{p}_{x2}}{D_1} \end{aligned}$$

being $D_1 \triangleq (\hat{p}_{x2}-\hat{p}_{x1})^2 + (\hat{p}_{y2}-\hat{p}_{y1})^2$ and $D_2 \triangleq D_1 + (\hat{p}_{z2}-\hat{p}_{z1})^2$.

The ${}^E\mathbf{e}$ error is computed from linearized (9) and (10) as:

$$ {}^E\mathbf{e} = F_1 \cdot {}^{P1}\mathbf{e} + F_2 \cdot {}^{P2}\mathbf{e} \tag{11}$$

where:

$$ F1 \triangleq J_{2\oplus}{}^{-1}\{{}^W\hat{\mathbf{x}}_E, 0\} \cdot F_1^* \cdot J_{2\oplus}\{{}^W\hat{\mathbf{x}}_{P1}, 0\}$$

$$ F2 \triangleq J_{2\oplus}{}^{-1}\{{}^W\hat{\mathbf{x}}_E, 0\} \cdot F_2^* \cdot J_{2\oplus}\{{}^W\hat{\mathbf{x}}_{P2}, 0\}$$

The covariance matrix of ${}^E\mathbf{e}$ is:

$$ Cov({}^E\mathbf{e}) = F_1 \cdot Cov({}^{P1}\mathbf{e}) \cdot F_1^T + F_2 \cdot Cov({}^{P2}\mathbf{e}) \cdot F_2^T$$

The methods outlined in this paragraph are used to get a initial estimate or to reduce uncertainty if we have previous estimates. In this case extended Kalman filter can be applied (Tardós, 1991).

With the sensed features, we compute some constraints (such as edge length, angle between planes, angle between edges, distance between parallel planes, distance between parallel edges) to locate and recognize the objets in the scene. These are calculated taking explicitly into account the feature uncertainty. Details about it can be found in Montano (1991b).

CONCLUSIONS

To observe geometric features, active sensing strategies using proximity sensors carried by the robot have been presented. When using active sensing, we try to exploit the discriminant power of features, that may be occluded to some sensors.

We have proposed a probabilistic model to represent geometric features. The 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. This is an homogeneous representation for the presented features. The measurement device and the robot location errors have been explicitly considered in the sensor model.

Some basic methods to acquire geometric features with proximity sensors have been exposed. They use non-contact guarded and compliant motions, which allow to achieve this goal. In other works, we use geometric constraints computed from the sensed uncertain features, to facilitate the recognition process.

Active sensing strategies with camera-in-hand and colaborating with proximity sensor will be dealt in future works.

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

Ellis R.E. (1987). Acquiring Tactile Data for the Recognition of Planar Objects. *IEEE Int. Conf. on Robotics and Automation*, 1799-1805.

Grimson W.E.L. (1990). *Object recognition by computer. The role of geometric constraints.* The M.I.T. Press.

Lee S., Hahn H.S. (1990). An Optimal Sensory Strategy of a Proximity Sensor System for Recognition and Localization of Polyhedral Objects. *IEEE Int. Conf. on Robotics and Automation*, 1666-1671.

Mason M. (1982). Compliance and Force Control for Computer Controlled Manipulators. In *Robot Motion: Planning and Control*, Ed. M. Brady et al., The M.I.T. Press, pp.373-404.

Montano L. (1987). *APRIL: Un Sistema Evolucionado de Programación de Robots.* PhD Thesis Dpto. de Ingeniería Eléctrica e Informática, University of Zaragoza, Spain.

Montano L, Sagüés C. (1991a). Non-Contact Compliant Robot Motions: Dynamic Behaviour and Application to Feature Localization. *IMACS-MCTS Symposium*, Lille, France.

Montano L., Sagüés C. (1991b). Active Sensing Using Proximity Sensors for Object Recognition and Localization. *Fifth International Conference on Advanced Robotics (ICAR'91)*, Pisa,Italy.

Paul R. P. (1981). *Robot Manipulators: Mathematics, Programming, and Control.*The M.I.T. Press, Cambridge, Mass.

Smith R., Self M., Cheeseman P. (1988). Estimating Uncertain Spatial Relationships in Robotics. In *Uncertainty in Artificial Intelligence 2*, Ed. J.F. Lemmer, L.N. Kanal, pp. 435-461.

Tardós J.D. (1991). *Integración multisensorial para reconocimiento y localización de objetos en robótica.* PhD Thesis Dpto. de Ingeniería Eléctrica e Informática, University of Zaragoza, Spain.

APPENDIX. Relating errors between frames

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

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

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

$$ \begin{aligned} {}^{F1}\mathbf{x}_{F2} &= (-{}^{F1}\mathbf{e}) \oplus {}^{F1}\hat{\mathbf{x}}_{W} \oplus {}^{W}\hat{\mathbf{x}}_{F2} \oplus {}^{F2}\mathbf{e} \\ &= ({}^{F1}\hat{\mathbf{x}}_{W} \oplus {}^{W}\hat{\mathbf{x}}_{F2}) \oplus ({}^{F2}\mathbf{e} - {}^{F2}J_{F1}{}^{F1}\mathbf{e}) \\ &= {}^{F1}\hat{\mathbf{x}}_{F2} \oplus {}^{F2}\mathbf{e}^{F1} \end{aligned} \tag{12} $$

If we have the transformation between frames expressed as:

$$ {}^{F1}\mathbf{x}_{F2} \simeq {}^{F1}\hat{\mathbf{x}}_{F2} + {}^{F2}\varepsilon^{F1} \tag{13} $$

The relation between ${}^{F2}\mathbf{e}^{F1}$ and ${}^{F2}\varepsilon^{F1}$ is obtained linearizing equation (12) in $\{{}^{F1}\hat{\mathbf{x}}_{F2}, {}^{F2}\hat{\mathbf{e}}^{F1}\}$:

$$ \begin{aligned} {}^{F1}\mathbf{x}_{F2} \simeq\ & {}^{F1}\hat{\mathbf{x}}_{F2} + \\ & + J_{1\oplus}\{{}^{F1}\hat{\mathbf{x}}_{F2}, {}^{F2}\hat{\mathbf{e}}^{F1}\} \cdot ({}^{F1}\hat{\mathbf{x}}_{F2} - {}^{F1}\hat{\mathbf{x}}_{F2}) \\ & + J_{2\oplus}\{{}^{F1}\hat{\mathbf{x}}_{F2}, {}^{F2}\hat{\mathbf{e}}^{F1}\} \cdot ({}^{F2}\mathbf{e}^{F1} - {}^{F2}\hat{\mathbf{e}}^{F1}) \\ & = {}^{F1}\hat{\mathbf{x}}_{F2} + J_{2\oplus}\{{}^{F1}\hat{\mathbf{x}}_{F2}, 0\} \cdot {}^{F2}\mathbf{e}^{F1} \end{aligned} \tag{14} $$

being ${}^{F2}\hat{\mathbf{e}}^{F1} = 0$, and $J_{1\oplus}\{\hat{\mathbf{x}}, \hat{\mathbf{e}}\}$ and $J_{2\oplus}\{\hat{\mathbf{x}}, \hat{\mathbf{e}}\}$ the composition jacobians with respect to the first and the second components, as defined in Tardós (1991). Equating (13) and (14) we reach to:

$$ {}^{F2}\varepsilon^{F1} = {}^{F1}\mathbf{x}_{F2} - {}^{F1}\hat{\mathbf{x}}_{F2} \simeq J_{2\oplus}\{{}^{F1}\hat{\mathbf{x}}_{F2}, 0\} \cdot {}^{F2}\mathbf{e}^{F1} $$