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

Internal Report: 1992-V05

Locating 3D Features with a Camera in Hand

C. Sagüés C., Montano L., Guerrero J.J.

If you want to cite this report, please use the following reference instead:
Locating 3D Features with a Camera in Hand, Sagüés C., Montano L., Guerrero J.J., *23rd International Symposium on Industrial Robots - ISBN 84-604-3652-7*, pages 423-428, October 1992.

This work was partially supported by project ROB91-0949 of the Comisión Interministerial de Ciencia y Tecnología (CICYT) and by project IT-5/90 of the Diputación General de Aragón (CONAI).

LOCATING 3D FEATURES WITH A CAMERA IN HAND

Sagüés C., Montano L., Guerrero J.J.

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

Abstract. This paper presents a procedure to obtain the 3D localization of straight edges from images taken by a camera-in-hand of a robot. Although location of polyhedral objects can be obtained from points, there are some advantages when using edges. We present two methods to compute the localization of edges, directly using the information of their projections in the image. One of them requires the identification of a tip and five degrees of freedom of the object are determined. The other do not require point identification, but only four degrees of freedom are obtained.

Keywords. Robot, active sensing, camera motion, edge localization.

1.- INTRODUCTION

Robotic system performance increases when it is provided with perceptual capabilities. Different kinds of sensors have been used to recognize and locate objects in the workspace: vision, range, and tactile sensors are some examples.

The importance of vision sensors has been broadly recognized for robotic tasks, mainly because of the amount of information which they provide about the objects in the scene. To obtain 3D location of features, at least, two images are required, which can be taken by stereo vision or by a mobile camera.

In active perception for recognition and localization, sensors are moved to get information from the scene. In some works, the sensors are moved to get the 3D location of the features. In others, the sensor is driven to gather information of some relevant feature to recognize the object. Motion allows to modify the point of view of the sensors, acquiring data from optimal directions of sensing. This possibility is more important when robot works in unstructured environments.

When a mobile camera is used, optical flow of the features in the consecutive images can be analyzed [Horn 86], [Waxman 85] or correspondence between features can be searched [Matthies 87b], [Broida 86], [Tsai 84].

The named works deal with points. The use of edges in the image renders some advantages. So, they supply more information from the scene and the position and orientation of the object can be better determined; obtaining edges in noisy images is easier and more accurate than points; total overlappings and occlusions are more difficult when observing edges than when observing points.

[Matthies 87a] and [Crowley 90] obtain object location by tracking edge lines but in these works the information about the edge is obtained from their points. [Liu 88] works with edges and states an algorithm to obtain 3D motion of an object from 6 lines in three images. [Kim 87] deduces the 3D motion of an object from 3D lines obtained from depth maps.

In this paper we tackle the localization of geometric features of static objects in an unstructured environments, from a mobile vision system located on the hand of an industrial robot. We consider polyhedral objects built by vertices, edges and planar surfaces, represented by a frame attached to them. We use a probabilistic model to characterize the uncertain locations of the features, which is presented in §2. In §3 a sensor model based on this representation is outlined, in which camera and robot position errors are considered.

We focus this work on obtaining not only location of points but also on 3D edge location from the information of its own projection in the image. In §4 the equations to obtain the depth of a point and two methods to obtain the 3D location of edges from camera motion are presented. Information from points and edges in the images can be integrated.

In the methods proposed to obtain edge location, the displacement of its projection has to be tracked, but it is easier than tracking points, because it can be carried out although the edge partially disappears from the image.

When camera motion is used to deduce the location of features, it is necessary to specify the motions to be made. Most of the related works use simple translational motions. The exposed methods in this work do not restrict its application, and they can be used when translational and rotational motions are commanded. Evidently, the equations deduced are simplified when using independent translational motions.

2.- REPRESENTING 3-D OBJECT GEOMETRIC FEATURES

We deal with polyhedral objects and we consider the information obtained from points and edges. To represent the location of these features we use a probabilistic model in which a frame is attached to them. The location of the feature F in the world reference W is expressed by the location vector of the frame

$$^W\mathbf{x}_F = (p_x, p_y, p_z, \psi_x, \theta_y, \phi_z)^T$$

where (p_x, p_y, p_z) is its origin and $(\psi_x, \theta_y, \phi_z)$ are its 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. As other representations proposed in the literature, this is overparameterized. In the integration and recognition process we will remove those parameters associated to d.o.f., because they have not valuable information (symmetries). In [Tardós 92] a technique based on selection matrices to take the parameters just needed is proposed.

To represent the location uncertainty we use a differential location vector

$$^F\mathbf{e} = (\delta p_x, \delta p_y, \delta p_z, \delta\psi_x, \delta\theta_y, \delta\phi_z)^T$$

associated to the feature frame. 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 $^W\hat{\mathbf{x}}_F \stackrel{\triangle}{=} E\{^W\mathbf{x}_F\}$ is the estimated value of $^W\mathbf{x}_F$, $^F\mathbf{e}$ and $^W\mathbf{e}$ are differential vectors in F and W frames, $^F J_W$ and $^W J_F$ are the jacobians to relate the errors between the references [Paul 81], and $\oplus$ is the operator to compose location vectors [Smith 88].

With the probabilistic model used, the location of the feature is given by the estimated value of the location vector ($^W\hat{\mathbf{x}}_F$) and the location uncertainty, that is characterized by the estimated value of the differential vector ($^F\hat{\mathbf{e}} = E\{^F\mathbf{e}\} = 0$) and its covariance matrix ($Cov(^F\mathbf{e})$).

So, assuming the hypothesis of gaussian white noise, $^W\mathbf{x}_F$ is completely defined by $^W\hat{\mathbf{x}}_F$ and $Cov(^F\mathbf{e})$

3.- SENSOR MODEL

A sensor model, based on the probabilistic model presented in §2, is defined. It is intimately tied to the camera location on the robot hand. It relates the location vector of the feature in the camera reference $^C\mathbf{x}_F$ and the location vector of the feature in the world reference $^W\mathbf{x}_F$, taking into account the uncertainty involved. This location can be computed as:

$$\begin{aligned} ^W\mathbf{x}_F &= (^W\hat{\mathbf{x}}_R \oplus {}^R\mathbf{e}_r \oplus {}^R\mathbf{x}_C) \oplus (^C\mathbf{e}_d \oplus {}^C\hat{\mathbf{x}}_F) \\ &= (^W\hat{\mathbf{x}}_R \oplus {}^R\mathbf{x}_C \oplus {}^C\hat{\mathbf{x}}_F) \oplus \\ &\quad \oplus (^F J_R \, {}^R\mathbf{e}_r \oplus {}^F J_C \, {}^C\mathbf{e}_d) \\ &= {}^W\hat{\mathbf{x}}_F \oplus {}^F\mathbf{e}_s \end{aligned} \tag{1}$$

being:

$^W\hat{\mathbf{x}}_R$	end-effector location vector estimate with respect to the world reference
$^R\mathbf{e}_r$	robot error vector expressed in its own reference system
$^R\mathbf{x}_C$	camera location vector with respect to the end effector
$^C\mathbf{e}_d$	image error vector expressed in camera reference
$^C\hat{\mathbf{x}}_F$	estimated location vector of the feature with respect to the camera reference system
$^F J_R$	jacobian to transform the robot error between its reference and feature reference
$^F J_C$	jacobian to transform the image error between its reference and feature reference

Assuming that $^C\mathbf{e}_d$ and $^R\mathbf{e}_r$ are known, the total error $^F\mathbf{e}_s$ is characterized by its covariance matrix, computed from (1) as follows:

$$Cov(^F\mathbf{e}_s) = {}^F J_R \, Cov(^R\mathbf{e}_r) \, {}^F J_R^T + {}^F J_C \, Cov(^C\mathbf{e}_d) \, {}^F J_C^T$$

This covariance matrix has 6×6 dimension, but depending on the kind of features being observed (points or edges) some of its elements have no interpretation. So, for points, the error vector has the form:

$$^P\mathbf{e}_s = (\delta p_x, \delta p_y, \delta p_z, 0, 0, 0)^T$$

and for the edges:

$$^E\mathbf{e}_s = (0, \delta p_y, \delta p_z, 0, \delta\theta_y, \delta\phi_z)^T$$

The null elements correspond to intrinsic uncertainty of the features due to their symmetries [Tardós 92] and they do not need to be characterized.

In this work we only consider the computation of the estimated values of locations. We are now working in the characterization of these errors and results will be reflected in future works.

4.- RELATING 3D LOCATION OF FEATURES WITH CAMERA MOTION

In our sensorial system camera motion is used to obtain the location of 3D features such as points and edges. Camera motion brings about the displacement of the projection of these features in the image. Relating the measured displacements with the commanded motions, the location of the features can be obtained. In this section, expressions which relate the motion of the camera between two images and the feature location are deduced. The known motion allows to predict the projection of the feature in the following images, and so easily find correspondences between them. Using several images with some integration mechanism, the location uncertainty is reduced.

4.1.- Localization of Points

In this section, the expressions to relate camera motion with a point location are summarized.

Let (p_x, p_y, p_z) be the coordinates of a point in the space expressed in the camera reference (C), and (x, y) the image coordinates in the same reference system. As it is well known, using the central projection model, both points are related by the equations:

$$x = \frac{f\, p_x}{p_z} \quad , \qquad y = \frac{f\, p_y}{p_z} \tag{2}$$

where f is the focal length.

If the camera moves by a differential motion $(dx, dy, dz, d\psi_x, d\theta_y, d\phi_z)$, the expressions relating the displacement of the point in the image with the six components of the camera motion are [Waxman 85]:

$$\begin{aligned} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \frac{1}{p_z}\begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}\begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} + \\ &+ \begin{bmatrix} \frac{x\, y}{f} & \frac{-x^2-f^2}{f} & y \\ \frac{y^2+f^2}{f} & \frac{-x\, y}{f} & -x \end{bmatrix}\begin{bmatrix} d\psi_x \\ d\theta_y \\ d\phi_z \end{bmatrix} \end{aligned} \tag{3}$$

From (3) equations, the p_z coordinate of the origin of the reference system of the point can be obtained. As can be seen, with the model used, only the translational motions allow to obtain information about the depth in the scene. When the translational motions are composed, we have two equations in (3) which combined will allow to reduce the uncertainty in the p_z estimated coordinate. With p_z, the p_x and p_y coordinates are obtained from 2. Taking any orientation parameters $(\psi_x, \theta_y, \phi_z)$, the estimated location of a point in the camera reference is:

$${}^C\hat{\mathbf{x}}_P = (p_x, p_y, p_z, \psi_x, \theta_y, \phi_z)^T$$

Finally, the estimated location of the point in the world reference is:

$${}^W\hat{\mathbf{x}}_P = {}^W\hat{\mathbf{x}}_C \oplus {}^C\hat{\mathbf{x}}_P$$

4.2.- Localization of Edges

Although edge localization can be obtained from points, these can not always be identified. Besides, it can happen that those points disappear as the camera moves, or become occluded by overlappings. Additionally, correspondence of edges between images can be easier to extract in a noisy image than of points.

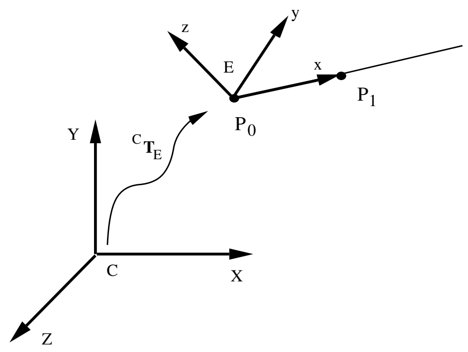

Figure 1: Edge reference system (E) and camera reference system (C)

We propose two methods to obtain 3D edge localization:

- *Edge with tip.* Edge localization is obtained from the identification of only one of their points (one tip) and the variation of the angle of its projection in successive images as the camera moves.
- *Edge without tips.* In this method, edge localization is obtained from any point of the edge in the images (by example the middle point) and the angle of its projection.

Edge with tip.

The equations to localizate the edge in the image are deduced from two points. After this deduction, we only must identify one of the tips of the edge to which the origin of its reference system is associated.

Let us suppose an edge with its reference system E, whose location can be expressed in the camera reference by a transformation ${}^CT_E$ (Fig. 1).

$${}^CT_E = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To localizate an edge in the scene it suffices with six of the twelve parameters of that transformation $(n_x, n_y, n_z, p_x, p_y, p_z)$, because the orientation of y and z parameters are irrelevant due to the symmetry of the edge around the x axis. If we take a point of this edge and we identify it, the elements (p_x, p_y, p_z) are obtained as from (2) and (3). The remainder elements are obtained from the angle of the projection in the image.

Let us to choose a point easily identifiable, for example a tip, whose coordinates are ${}^E\mathbf{P}_0 = (0, 0, 0, 1)^T$ and other point (${}^E\mathbf{P}_1$) at the end of a unit vector along the x axis of the reference system of the edge (Fig. 1) whose coordinates are ${}^E\mathbf{P}_1 = (1, 0, 0, 1)^T$.

The coordinates of the projection of ${}^E\mathbf{P}_0$ is obtained as:

$${}^I\mathbf{P}_0 = P_P\, {}^CT_E\, {}^E\mathbf{P}_0$$

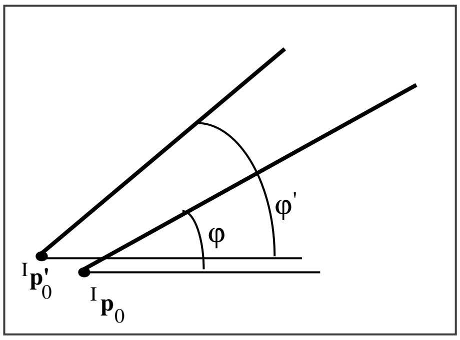

Figure 2: Projections and angles of the edge when the camera moves

being P_P the perspective projection matrix:

$$P_P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix}$$

The homogeneous coordinates of this projection point are:

$$^I\mathbf{P}_0 = \left[\frac{f\ p_x}{p_z}\ ,\ \frac{f\ p_y}{p_z}\ ,\ f\ ,1 \right]^T$$

The third component of this vector corresponds with z coordinate of the image plane. So the coordinates of the point in the image can be written as :

$$^I\mathbf{p}_0 = (x_0, y_0) = (\frac{f\ p_x}{p_z}\ ,\ \frac{f\ p_y}{p_z})$$

Similarly, it can be deduced the coordinates of $^E\mathbf{P}_1$ in the image

$$^I\mathbf{p}_1 = (x_1, y_1) = (f\ \frac{p_x + n_x}{p_z + n_z}\ ,\ f\ \frac{p_y + n_y}{p_z + n_z})$$

The angle of the line through those points $^I\mathbf{p}_0$ and $^I\mathbf{p}_1$ is (Fig. 2):

$$\varphi = tan^{-1}\ \frac{y_1 - y_0}{x_1 - x_0} = tan^{-1}\ \frac{p_z n_y - p_y n_z}{p_z n_x - p_x n_z} \tag{4}$$

When the camera moves, the coordinates of the points $^E\mathbf{P}_0$ y $^E\mathbf{P}_1$ expressed in the camera reference system change and their projections in the new image are:

$$\begin{aligned} ^I\mathbf{P}'_0 &= P_P\ \ dT^{-1}\ \ ^CT_E\ \ ^E\mathbf{P}_0 = P_P\ \ ^{C'}T_E\ \ ^E\mathbf{P}_0 \\ ^I\mathbf{P}'_1 &= P_P\ \ dT^{-1}\ \ ^CT_E\ \ ^E\mathbf{P}_1 = P_P\ \ ^{C'}T_E\ \ ^E\mathbf{P}_1 \end{aligned}$$

where dT is the differential transformation representing the camera displacement in C reference:

$$dT = \begin{bmatrix} 1 & -d\phi_z & d\theta_y & dx \\ d\phi_z & 1 & -d\psi_x & dy \\ -d\theta_y & d\psi_x & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} dR & \mathbf{dp} \\ 0 & 1 \end{bmatrix}$$

The transformation to express edge location in the new position of the camera is:

$$^{C'}T_E \ = \begin{bmatrix} n'_x & o'_x & a'_x & p'_x \\ n'_y & o'_y & a'_y & p'_y \\ n'_z & o'_z & a'_z & p'_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where its elements are:

$$\begin{aligned} n'_x &= n_x - d\phi_z n_y + d\theta_y n_z \\ n'_y &= d\phi_z n_x + n_y - d\psi_x n_z \\ n'_z &= -d\theta_y n_x + d\psi_x n_y + n_z \\ p'_x &= p_x - d\phi_z p_y + d\theta_y p_z + dx \\ p'_y &= d\phi_z p_x + p_y - d\psi_x p_z + dy \\ p'_z &= -\delta\theta_y p_x + \delta\psi_x p_y + p_z + \delta z \end{aligned} \tag{5}$$

Similarly than above, we obtain the coordinates of both points in the new image:

$$^I\mathbf{p}'_0 = (x'_0, y'_0) = \left(f\frac{p'_x}{p'_z}\ ,\ f\frac{p'_y}{p'_z} \right)$$

$$^I\mathbf{p}'_1 = (x'_1, y'_1) = \left(f\frac{p'_x + n'_x}{p'_z + n'_z}\ ,\ f\frac{p'_y + n'_y}{p'_z + n'_z} \right)$$

The angle of the line in the new image is (Fig. 2):

$$\varphi' = tan^{-1}\ \frac{p'_z n'_y - p'_y n'_z}{p'_z n'_x - p'_x n'_z} \tag{6}$$

The proposed method to localizate a 3D edge with an identified tip can be summarized in the following steps:

- Take the first image. Extract the $^I\mathbf{p}_0$ coordinates of the tip of the edge and the φ angle of its projection.
- Move the camera by dT
- Extract the $^I\mathbf{p}'_0$ coordinates of the same point and the φ' angle of its projection in this second image. The system must find the corresponding point in both images. It can use (3) equation to predict its location in the new image and, therefore, the search can be focused on a reduced window centered around the prediction.
- Compute $\mathbf{p} = (p_x, p_y, p_z)^T$ of the tip of the edge with (2) and (3) equations. These coordinates correspond with the origin of the reference system attached to the edge.
- From (4), (5) and (6) equations and applying the orthonormality condition, $n_x^2 + n_y^2 + n_z^2 = 1$, solve the system equations to obtain $\mathbf{n} = (n_x, n_y, n_z)^T$. When the motions are simple, it is easy to find a closed-form expression for $\mathbf{n}$. In other cases, if rotational motions are involved, the solution can be found by numerical procedures.
- Compute the location vector $^C\hat{\mathbf{x}}_E$ of the frame of the edge from $\mathbf{p}$ and $\mathbf{n}$ as follows:

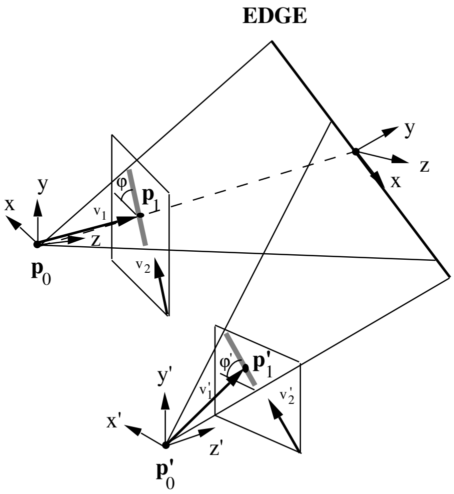

Figure 3: Edge without tips localization from two images

$$\begin{aligned} \phi_z &= atan2\ (n_y, n_x) \\ \theta_y &= atan2\ (-n_z, n_x cos\phi_z + n_y sin\phi_z) \end{aligned} \tag{7}$$

and ψ_x can take any value. Then, the estimated location is expressed as:

$$^C\hat{\mathbf{x}}_E = (p_x, p_y, p_z, \psi_x, \theta_y, \phi_z)^T$$

With this method, we directly work with the orientation of the projection of the edge in the image, but we have to identify one point on the edge to solve the equations.

Edge without tips

As the identification of one tip is not always possible (occlusions and overlappings), we propose an alternative method to obtain the localization of the edge. In this method it is not necessary to identify any points. The localization of the edge is computed from intersection of the projection planes of the edge in both images.

The method proposed can be summarized in the following steps:

- Take an image from the scene. Extract the angle of the projection of the edge (φ) and the coordinates of one of their points (by example the middle point) $^C\mathbf{p}_1 = (p_{x_1},\ p_{y_1},\ p_{z_1})^T$. The position of the optical centre of the camera $^C\mathbf{p}_0 = (0,\ 0,\ 0)^T$ is the origin of the reference system.

- Compute $\mathbf{v}_1$ and $\mathbf{v}_2$ vectors which belong to the projection plane of the edge (Fig. 3) as follows:

$$\begin{aligned} \mathbf{v}_1 &= (p_{x_1},\ p_{y_1},\ p_{z_1})^T \\ \mathbf{v}_2 &= (1,\ tan\ \varphi,\ 0)^T \end{aligned}$$

- Compute the unit vector normal to projection plane:

$$^C\mathbf{a}_1 = \frac{\mathbf{v}_1 \times \mathbf{v}_2}{||\mathbf{v}_1 \times \mathbf{v}_2||}$$

  where $\times$ is the cross vector and $||\ ||$ is the modulus of the vector.

- Move the camera dT. Extract the angle of the projection of the edge (φ') and one of their points $^{C'}\mathbf{p}'_1 = (p'_{x_1},\ p'_{y_1},\ p'_{z_1})^T$ and the position of the optical centre $^{C'}\mathbf{p}'_0 = (0,\ 0,\ 0)^T$ in the new reference of the camera.

- Compute $\mathbf{v}'_1$ and $\mathbf{v}'_2$ vectors belonging to the projection plane of the edge in this camera position as follows:

$$\begin{aligned} \mathbf{v}'_1 &= (p'_{x_1},\ p'_{y_1},\ p'_{z_1})^T \\ \mathbf{v}'_2 &= (1,\ tan\ \varphi',\ 0)^T \end{aligned}$$

- Compute the normal to the new projection plane:

$$^{C'}\mathbf{a}_2 = \frac{\mathbf{v}'_1 \times \mathbf{v}'_2}{||\mathbf{v}'_1 \times \mathbf{v}'_2||}$$

- Transform the vector ($^{C'}\mathbf{a}_2$) and the point ($^{C'}\mathbf{p}'_0$) to the C reference:

$$^C\mathbf{a}_2 = dR\ ^{C'}\mathbf{a}_2$$

$$^C\mathbf{p}'_0 = dR\ ^{C'}\mathbf{p}'_0 + \mathbf{dp}$$

- Obtain the direction of the x axis of the reference system of the edge, which is the direction of the common perpendicular to $^C\mathbf{a}_1$ and $^C\mathbf{a}_2$ belonging to both planes:

$$\mathbf{n} = \frac{^C\mathbf{a}_1 \times {}^C\mathbf{a}_2}{||^C\mathbf{a}_1 \times {}^C\mathbf{a}_2||}$$

- Compute a perpendicular plane to the plane built by $\mathbf{v}_1$ y $\mathbf{v}_2$ vectors passing through the point $^C\mathbf{p}_1$. The unit vector of its normal direction can be obtained as:

$$^C\mathbf{a}_3 = \frac{\mathbf{v}_1 \times {}^C\mathbf{a}_1}{||\mathbf{v}_1 \times {}^C\mathbf{a}_1||}$$

- Obtain a point $\mathbf{p}$ of the edge to locate its reference system, as intersection of three planes, by solving the equations:

$$\begin{aligned} {}^C\mathbf{a}_1{}^T\ (\mathbf{p} - {}^C\mathbf{p}_0) &= 0 \\ {}^C\mathbf{a}_2{}^T\ (\mathbf{p} - {}^C\mathbf{p}'_0) &= 0 \\ {}^C\mathbf{a}_3{}^T\ (\mathbf{p} - {}^C\mathbf{p}_0) &= 0 \end{aligned}$$

- Compute the estimated location vector $^C\hat{\mathbf{x}}_E$ from $\mathbf{p}$ and $\mathbf{n}$ as in (7).

As can be seen this procedure is simpler than the exposed above, because is not necessary to find correspondence between points in the images.

With the first method (*edge with tip*), five d.o.f. of the edge are determined, because we compute its direction and one of its points. With the second method (*edge without tip*) only four d.o.f. are fixed corresponding with the d.o.f. of a 3D straight line. Therefore the first method contributes to a lower uncertainty in the object location at which the edge belongs. However, in this case it is necessary to identify a point which can be difficult.

5.- CONCLUSIONS

In this work we have presented a procedure to obtain the 3D localization of geometric features as points and edges, gathered by a camera placed in the hand of a robot. This camera belongs to an active multisensorial system been made for the localization and recognition of objects in the scene.

The aim of this work has been to present two alternative methods to localizate edges when objects can appear overlapped or the images obtained are noisy. These methods are different from those which work with points and we have proposed to work directly with information of straight lines.

In the first method, the 3D location of an edge is obtained from an identified point and from the change of its angle in the image. The method allows to fix five d.o.f., but it is necessary to find a correspondent point in the images. The search can be simplified predicting the projection of the point as camera moves, since this motion is known.

The second method, allows to locate the edge without searching correspondences between points in the images. However only four d.o.f. are fixed, and therefore, its contribution to locate the object at which the edge belongs is lower than with the first method.

As camera motions can be controlled, these can be made as short as desired for easier finding the corresponding features in the sequence of images or not loosing the features of interest from the image. As short motions involve poor resolution, the data of several images can be fused with an integration mechanism to reduce uncertainty in the feature and object localization.

We have computed locations without considering the uncertainty in the sensed features. In future works we will tackle the uncertainty taking into account the camera and robot motion errors, and using the probabilistic model we have presented.

ACKNOWLEDGEMENTS

This work was partially supported by project ROB91-0949 of the Comisión Interministerial de Ciencia y Tecnología (CICYT) and by project IT-5/90 of the Diputación General de Aragón (CONAI).

REFERENCES

References

[Broida 86] T.J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):90–99, 1986.

[Crowley 90] J.L. Crowley and P. Stelmaszyk. Measurement and integration of 3-D structures by tracking edge lines. In *First European Conference on Computer Vision*, pages 269–280, Antibes, France, 1990.

[Horn 86] B.P.K. Horn. *Robot Vision*. MIT Press, Cambridge, Mass., 1986.

[Kim 87] Y.C. Kim and J.K. Aggarwal. Determining object motion in a sequence of stereo images. *IEEE Journal of Robotics and Automation*, 3(6):599–614, 1987.

[Liu 88] Y. Liu and T.S.Huang. Estimation of rigid body motion using straight line correspondences. *Computer Vision, Graphics And Image Processing*, (43):37–52, 1988.

[Matthies 87a] L. Matthies and T. Kanade. The cycle of uncertainty and constraint in robot perception. In R.C. Bolles and B. Roth, editors, *Robotics Research: The Fourth International Symposium*, pages 327–335. MIT Press, 1987.

[Matthies 87b] L. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. Technical Report CMU-CS-87-185, Carnegie Mellon University, Pittsburgh, PA 15213, 1987.

[Paul 81] R.P. Paul. *Robot Manipulators: Mathematics, Programming, and Control*. MIT Press, Cambridge, Mass., 1981.

[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.

[Tardós 92] 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.

[Tsai 84] R.Y. Tsai and T.S. Huang. Uniqueness and estimation of three-dimensional motion paameters of rigid objects with curved surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(1):13–27, 1984.

[Waxman 85] A.M. Waxman and K. Wohn. Contour evolution, neighborhood deformation, and global image flow: Planar surfaces in motion. *Int. J. of Robotics Research*, 4(3):95–108, 1985.