A Robust approach to control robot manipulators by fusing visual and force information

J. Pomares, G. J. García, F. Torres

Department of Physics, System Engineering and Signal Theory
University of Alicante
P. O. Box 99, 03080 Alicante, Spain
{jpomares, gjgg, Fernando.Torres}@ua.es
Tel: +34 965903400. Fax: +34 965909750

Abstract— In this paper, a method to combine visual and force information using the information obtained from the movement flow-based visual servoing system is proposed. This method allows not only to achieve a given desired position but also to specify the trajectory that the robot will follow from its initial position to its final one in the 3D space. This paper also extends the visual servoing system in order to increase the robustness when errors in the camera calibration parameters appear. Experiments using an eye-in-hand robotic system demonstrate the correct behaviour when important errors exist in the camera intrinsic parameters. After the description of this strategy, we then describe its application to an insertion task to be performed by the robotic system in which the joint use of visual and force information is required. To combine both sensorial systems, a position-based impedance-control system is implemented, which modifies the trajectory generated by the visual system depending on the robot's interaction with its setting. This modification is performed without knowledge of the exact camera calibration parameters. Furthermore, the visual-force approach based on impedance control does not require having previous knowledge about the contact geometry.

Categories (7), (8).

1. INTRODUCTION

Actually, visual servoing systems are a well-known approach to guide a robot using image information. Typically, visual servoing systems are position-based and image-based classified [1]. Position-based visual servoing requires the computation of a 3D Cartesian error for which a perfect CAD-model of the object and a calibrated camera are necessary. These types of systems are very sensitive to modeling errors and noise perturbations. In image-based visual servoing the error is directly measured in the image. This approach ensures the robustness with respect to modeling errors, but generally an inadequate camera motion in the 3D Cartesian space is obtained [2]. On the other hand, it is well known that image-based visual servoing is locally stable. This nice property ensures the correct convergence if the desired configuration is sufficiently near to the current one. Consequently, in these systems the desired position is indicated but the trajectory that should be followed to arrive at such a position is not. To perform the tracking of a given trajectory using a robot eye-in-hand camera system the approach called movement flow-based visual servoing have been used [3]. Movement flow-based visual servoing is employed to track previously generated trajectories in the image space. Doing so, the visual servoing system allows the robot to develop the tracking of the desired 3D trajectory in a non time dependent way. This last aspect allows its application when visual and force information are required to be used jointly to develop the task. In this paper, several aspects of this approach are extended or improved when it is combined with force sensor information. In the paper, the system robustness is demonstrated when errors in the camera intrinsic parameters occur. To demonstrate the correct behaviour, the system is applied to an insertion task that requires the combined force-position control.

The use of multisensorial systems increases the flexibility and the robustness of any manipulation task. To do so, several sensor fusion systems have been proposed. Within these approaches we can mention the works of Nandi and Mitra [4] which carries out the fusion by

minimizing uncertainties in robotic manipulation. However, in order to fuse the visual and force information, the fact that the two sensors measure different physical phenomena of different natures must be taken into consideration. That is why the classical sensor fusion methods cannot be used and theories of force-position hybrid control [5] have been applied. We should also mention other techniques, such as the concept of "resolvability" [6], which affords a measurement of each sensors' ability to resolve the movement; or the works of Zhou et al. [7] which focus on micro-manipulation; or Von Collani et al.[8] who employ a neuro-fuzzy solution for the integration. Nevertheless, a new approach to fuse visual and force information based on impedance control is proposed in this paper. This method presents several advantages over the previous visual-force control systems (and furthermore over our previous works [3]). Within these advantages we can mention the possibility of combining both sensors in situations in which the control actions are contradictory. To solve this problem the system determines the more adequate sensor to control the task, guaranteeing the system robustness with respect to camera calibration errors. In our previous works [3] we have fused the control actions from the force and visual controllers when previous knowledge about the workspace is provide. However, considering a realistic application with unknown constraint location it is not possible to previously know the contact geometry. In this case, the use of the proposed impedance control approach is more suitable. Unpredicted contacts can occur when the robot gets close to the workspace. In this case, the force controller must provide compliance to limit the interaction forces and guide the manipulator to the final location. To do so, an approach to fuse visual and force information using the impedance approach is proposed which is adequate in non-structured environments and when new contacts appears.

In this article, the information from the movement flow is employed to combine the two different sensorial information. The concept of visual impedance [9] is used together with our previous experiences in fusing multi-sensorial systems composed of visual and force [3].

Another important contribution of this paper is the processing of information in order to guarantee the coherence between the force and the visual information. To do so, a new method described in Section 5 is proposed. This section presents a method for the image trajectory modification which is demonstrated to be robust when errors in the camera intrinsic parameters appear.

This paper is organized as follows: Section 2 describes the main characteristics of the tracked trajectory and the notation used; Section 3 shows the main concepts concerning the movement flow-based visual servoing system; in Section 4, the formulation of the visual-force control system used is detailed; Section 5 describes the modification of the desired image trajectory from the interaction forces obtained during the task; in Section 6, experimental results, using an eye-in-hand camera system, confirm the validity of the proposed control scheme. The final section presents the main conclusions arrived at.

2. NOTATION

In this paper, the presence of a planner, which provides the robot with the 3D trajectory, $\gamma(t)$, to be tracked (i.e. the desired 3D trajectory of the camera at the end-effector), is assumed. For this study we have employed planners developed in our previous works [10][11].

From $\gamma(t)$ a sampling of the trajectory is done. Consequently, a sequence of N discrete values of $\gamma(t)$, each of them representing an intermediate camera position $\tau = \left\{ {}^{k}\gamma/k \in 1...N \right\}$ is obtained. ${}^{k}\mathbf{P}_{i} \, / \, i \in 1...M$ are the 3D coordinates (with respect to the camera coordinate frame) of the points extracted by the camera in position ${}^{k}\boldsymbol{\gamma}$. The projective coordinates in the image of a given point ${}^{k}\mathbf{P}_{i}$ can be obtained as ${}^{k}\boldsymbol{f}_{i} = \mathbf{A} \, {}^{k}\mathbf{P}_{i}$, where $\mathbf{A}$ is a matrix having the camera internal parameters.

$$A = \begin{bmatrix} f \cdot p_u & -f \cdot p_u \cdot \cot(\theta) & u_0 \\ 0 & f \cdot p_v / \sin(\theta) & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where u_0 and v_0 are the pixel coordinates of the principal point, f is the focal length, p_u and p_v are the magnifications in the u and v directions respectively, and θ is the angle between these axes.

From τ the discreet trajectory of the features in the image is obtained $S = \left\{ {}^k s / k \in 1...N \right\}$, with ${}^k s$ being the set of M points or features observed by the camera at instant k, ${}^k s = \left\{ {}^k f_i / i \in 1...M \right\}$. Figure 1 shows an example of 3D and image trajectory in order to illustrate the previous mentioned notation.
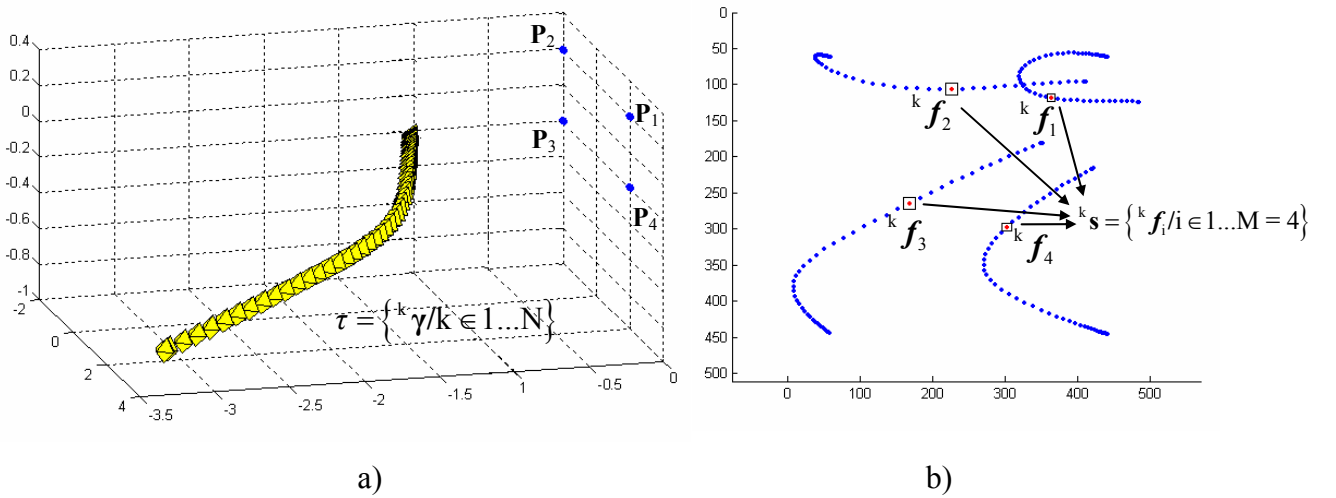

a)   b)

Figure 1. Notation employed a) 3D trajectory to be tracked, $\tau = \left\{ {}^k \gamma / k \in 1...N \right\}$. b) Image trajectory to be tracked $S = \left\{ {}^k s / k \in 1...N \right\}$.

3. MOVEMENT FLOW BASED VISUAL SERVOING

In this section, a review of the main aspects about the movement flow-based visual servoing systems is presented. The movement flow is a vector field that converges towards the desired trajectory (see in Figure 2.a a detail of the movement flow obtained for a given trajectory).

The value of the movement flow at each point of the desired image trajectory is a vector tangent to the desired trajectory. However, the values outside the trajectory aim to decrease the tracking error. Consequently, movement flow is a vector field that indicates the direction in which the desired features to be used by an image-based visual servoing system must be located. This way, it allows the tracking of the trajectory. Hence, considering image-based control, the velocity applied to the robot with respect to the camera coordinate frame will be:

$$v_M^C = -\lambda_M \cdot \hat{J}_f^+ \cdot e_f \tag{1}$$

where λ_M is the gain of the proportional controller; $\hat{J}_f^+$ is the estimated pseudo-inverse of the interaction matrix [1], $e_f = s - s_d$, $s = [f_1, f_2, ..., f_M]^T$ are the features extracted from the image, $s_d = [f_1 + m_1 \Phi_1(f_1), f_2 + m_2 \Phi_2(f_2), ..., f_M + m_M \Phi_M(f_M)]^T$, Φ_i is the movement flow for the feature i and $m = \{m_1, m_2, ..., m_M\}$ determines the progression speed.

Y axis (pix)

X axis (pix)

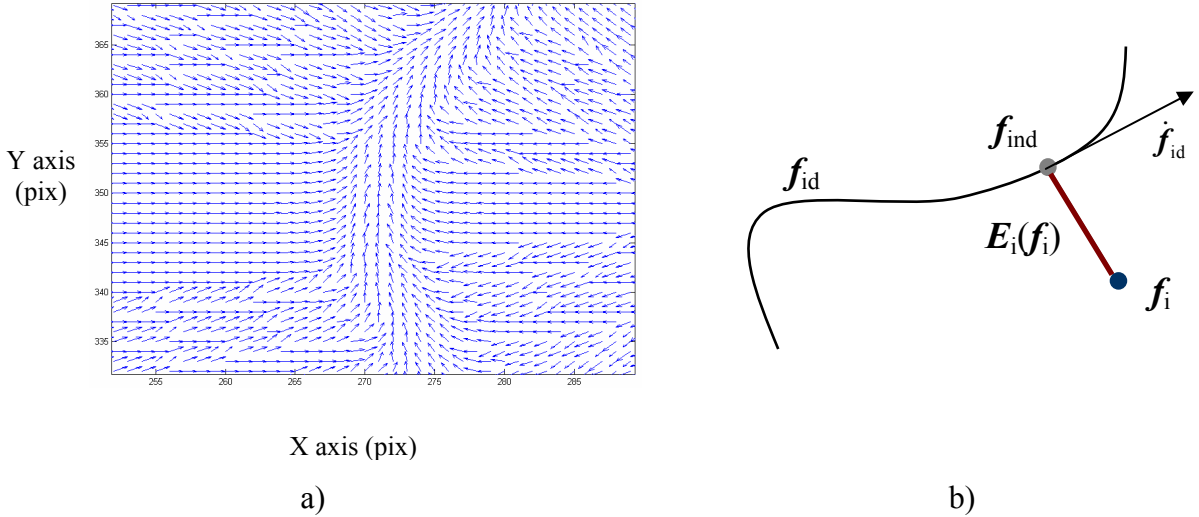a)   b)

Figure 2. a) Detail of the movement flow obtained for the feature f_1 of Figure 1.b. b) Main notation used for the definition of the movement flow.

Considering that f_i are the coordinates of the feature i in the image and that the coordinates of the nearest point to it in the desired trajectory are $f_{ind} = (f_{xind}, f_{yind})$, the error vector $E_i(f_i) = (E_{xi}, E_{yi})$ where $E_{xi} = (f_{xi} - f_{xind})$ and $E_{yi} = (f_{yi} - f_{yind})$ is defined (See Figure 2.b). From this

error the potential function $\mathbf{U}_i:\mathfrak{R}^n \to \mathfrak{R}$ is computed. Based on these functions, the movement flow for the feature i, Φ_i, is defined as a linear combination of two terms:

$$\Phi_i(\boldsymbol{f}_i) = \begin{pmatrix} \Phi_{xi}(\boldsymbol{f}_i) \\ \Phi_{yi}(\boldsymbol{f}_i) \end{pmatrix} = G_1(\boldsymbol{f}_i) \cdot \begin{pmatrix} \dot{f}_{xid} \\ \dot{f}_{yid} \end{pmatrix} - G_2(\boldsymbol{f}_i) \cdot \begin{pmatrix} \dfrac{\partial \mathbf{U}_i}{\partial E_{xi}} \\ \dfrac{\partial \mathbf{U}_i}{\partial E_{yi}} \end{pmatrix} \tag{2}$$

where G_1, G_2: $\mathfrak{I} \to \mathfrak{R}^+$ are weight functions so that $G_1 + G_2 = 1$. The first term in (2), $\dot{\boldsymbol{f}}_{id}$, is a tangent vector to the desired trajectory in $\boldsymbol{f}_{ind}$. Therefore, G_1 controls the progression of the trajectory in the image. The second term in Equation (2) is employed to reduce the tracking error. G_2 controls the strength of the gradient field. The value of the weight functions G_1 and G_2, the potential function, $\mathbf{U}_i$, employed in (2) and more details about the movement flow-based visual servoing system can be seen in [3].

4   FUSING FORCE AND VISUAL INFORMATION

So far, most of the applications developed to combine visual and force information employ hybrid control [5][6]. In such a case, the workspace must be precisely known, so that a division of the directions controlled by force and those controlled by position (i.e., using a visual servoing system) is first carried out. With a view to increasing the versatility of the system and improving its response to the uncertainties that arise during a manipulation task, a special visual impedance control system has been developed. It uses the information obtained form the movement flow-based visual servoing technique. This system has an external visual feedback, in contrast to others approaches [12] which have the vision and force systems at the same level of the control hierarchy. The proposed approach also provides important improvements with respect to our previous visual-force control systems [3]. This previous method can only be used in applications in which the robot must maintain constant contact

with a given surface. However, this method is not adequate in situations in which new contacts appear. In this paper, the impedance visual-force control system allows the correct behaviour in situations in which new contacts occur during the task. Furthermore, in order to apply the proposed approach in more realistic applications, a method which does not require previous knowledge about the contact geometry has been defined. This aspect allows the application of the method to unstructured environments such as insertion tasks in which the system has not previous information about the geometry of the workspace.

The objective of the impedance control is to carry out the combined control of the movement of the robot and its interaction force. It can be stated that the case described in this article is active control, in which force feedback is used to control the movements of the joints. The following impedance equation enforces an equivalent mass-damper-spring behaviour for the pose displacement when the end-effector exerts a force F on the environment:

$$F = \mathbf{I}\Delta\ddot{x}_{vc} + \mathbf{D}\Delta\dot{x}_{vc} + \mathbf{K}\Delta x_{vc} \tag{3}$$

where x_c is the current end-effector pose, x_v is the reference trajectory, $\Delta x_{vc} = x_v - x_c$, $\mathbf{I} \in \mathfrak{R}^{n \times n}$ is the inertial matrix, $\mathbf{D} \in \mathfrak{R}^{n \times n}$ is damping matrix and $\mathbf{K} \in \mathfrak{R}^{n \times n}$ is the stiffness. They are diagonal matrices and characterize the desired impedance function.

To implement this controller, an external visual feedback has been introduced in the impedance control scheme with an internal movement feedback. To implement the controller, we have chosen a position-based impedance control system called accommodation control [13] in which the desired impedance is limited to pure damping $\mathbf{D}$. In this case:

$$F = \mathbf{D}\Delta\dot{x}_{vc} \tag{4}$$

Therefore, the control law obtained will be:

$$\dot{x}_c = \dot{x}_v - \mathbf{D}^{-1}F \tag{5}$$

where the term $\dot{x}_v$ is obtained by using the movement flow-based visual servoing system:

$$\dot{x}_v = -k \cdot \mathbf{C}_v \cdot \hat{\mathbf{J}}_f^+ \cdot \left(\begin{bmatrix} f_1 \\ f_2 \\ ... \\ f_M \end{bmatrix} - \begin{bmatrix} f_1 + m_1 \Phi_1 (f_1) \\ f_2 + m_2 \Phi_2 (f_2) \\ ... \\ f_M + m_M \Phi_M (f_M) \end{bmatrix} \right) \tag{6}$$

where the velocity is mapped from the camera coordinate frame to the robot end-effector frame by using the matrix $\mathbf{C}_v$. As it is shown in [3] from the movement flow-based visual servoing system the image error decreases exponentially. Therefore, the first term in (5) also decreases exponentially. In order to demonstrate the system stability it is necessary to show that the second term in (5) also decreases exponentially. To do so, we firstly consider $e_F = \mathbf{D}^{-1} F$. Therefore:

$$\dot{e}_F = \mathbf{D}^{-1} \dot{F} = \mathbf{D}^{-1} \mathbf{L}_F \dot{x}_c \tag{7}$$

where $\mathbf{L}_F = \dfrac{\partial F}{\partial x_c}$. From (7) it is possible to obtain $\dot{x}_c = \left(\mathbf{D}^{-1} \mathbf{L}_F \right)^{-1} \dot{e}_F$ and by imposing an exponential decrease of the error function $\dot{e}_F = -\lambda e_F$ $(\lambda > 0)$ the following expression is obtained:

$$\dot{x}_c = -\lambda \left(\mathbf{D}^{-1} \mathbf{L}_F \right)^{-1} e_F \tag{8}$$

Therefore, in order to obtain an exponential decrease it is necessary that $\mathbf{D}^{-1} \mathbf{L}_F > 0$. In [14] this term is demonstrated to be greater than zero.

In Figure 3 the proposed control impedance scheme is shown. In this figure the inverse kinematics component implements a conventional pseudo inverse algorithm, $\dot{q}_c = \mathbf{J}_q^+ \dot{x}_c$, where $\dot{q}_c$ are the corresponding joint velocities and $\mathbf{J}_q^+$ is the pseudo inverse of the Jacobian matrix.
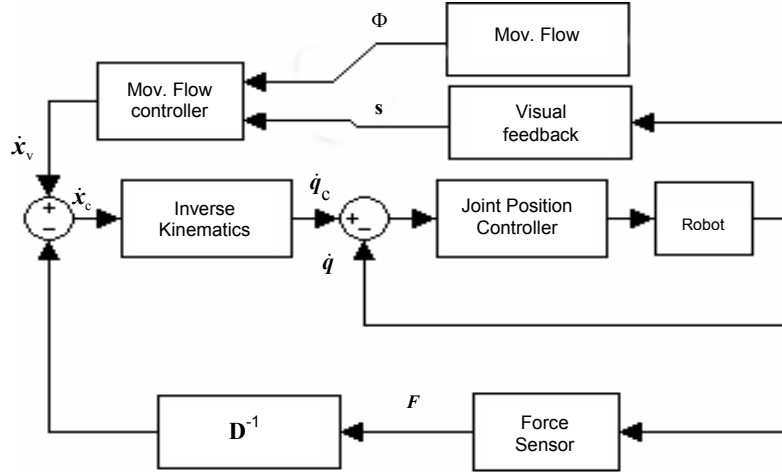
Figure 3. Impedance control scheme.

Therefore, by defining the desired damping **D**, the data from both sensors are combined as shown in Equation (5). This way the trajectory generated by the movement flow-based visual servoing system is corrected according to the robot's interaction.

5 IMAGE TRAJECTORY MODIFICATION FROM INTERACTION FORCES

So far, approaches to fuse visual and force information do not consider the possibility of both sensors providing contradictory information at a given moment of the task (i.e., the visual servoing system establishes a movement direction that is impossible according to the interaction information obtained from the force sensor).

To assure that a given task in which it is required an interaction with the setting is correctly developed, the system must carry out a variation of the trajectory in the image, depending on the spatial restrictions imposed by the interaction forces. This aspect has been addressed in our previous work [3]. However, in this paper an important improvement of the presented method is described. This improvement consists of the modification of the image trajectory assuring the robustness when errors in the camera intrinsic parameters appear. To do so, given a collision with the workspace and having recognized the normal vector n of the contact surface [3], the transformation that the camera must undergo to fulfil the spatial

restrictions is determined. This transformation, given by a rotation matrix **R** and a translation *t*, is calculated so that it represents the nearest direction to the one obtained from the movement flow-based visual servoing, and which is contained in the plane of the surface.

Therefore, to guarantee that visual information will be coherent with that obtained from the force sensor, the image trajectory must be modified, so that the new trajectory is the one obtained from the previous mentioned transformation. To do so, we must introduce the concept of projective homography matrix **G**. This matrix relates the points viewed from a camera configuration with the same points observed from another position of the camera. If we compute matrix **G** at each iteration (i.e, each time an image is captured) according with the transformation obtained after a collision is detected (through the rotation matrix **R** and the translation *t*), we will be able to obtain the new trajectory in the image space.
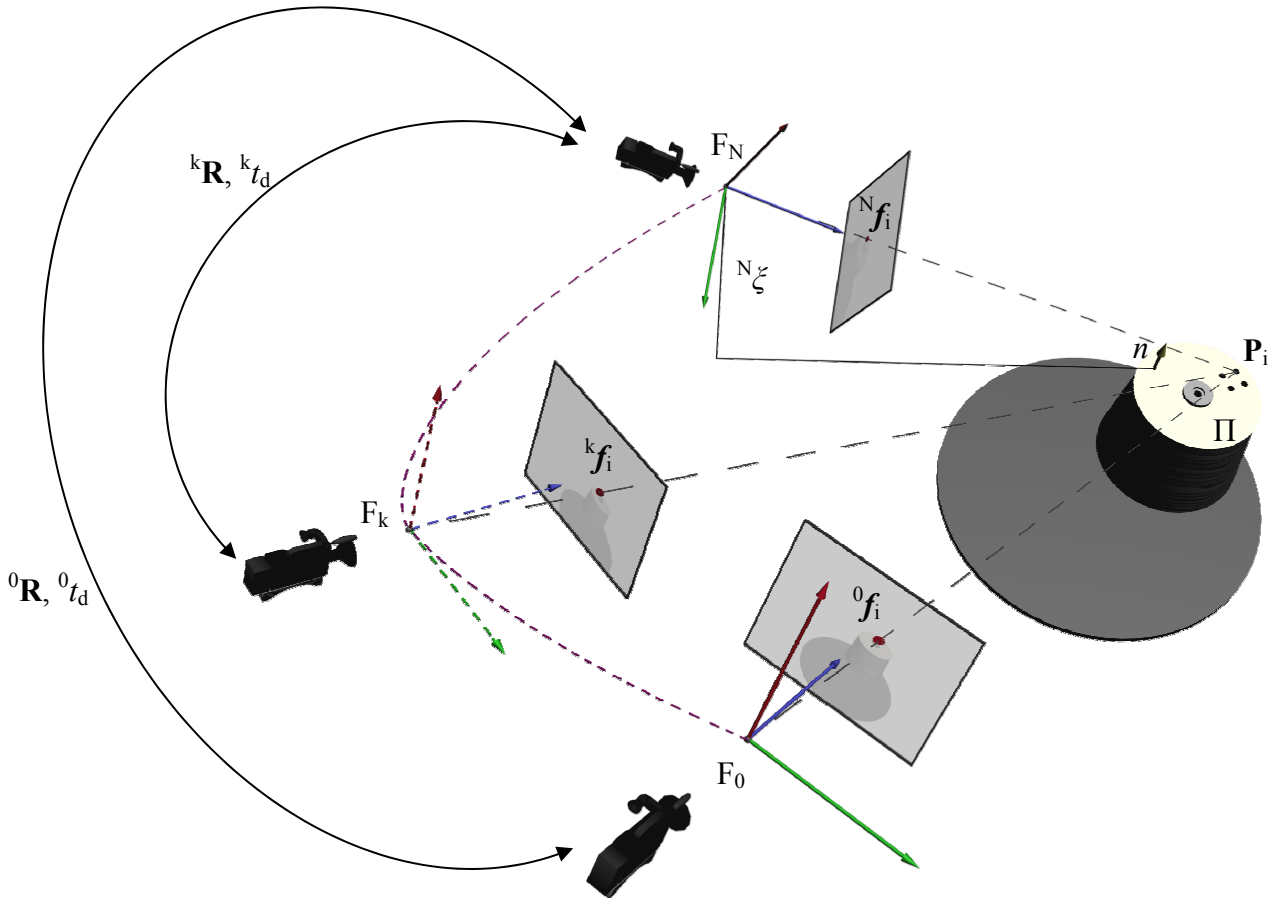
Figure 4. Scheme of the motion reconstruction.

Firstly, we call Π the contact surface plane. Once the collision is detected, considering **P**$_i$ a 3D point observed by the camera (representing a feature extracted from the object), the same point in the image space is f_i. ^kf_i represents the position of the feature in the image captured in an intermediate position k of the camera. In the target image (which corresponds with the image obtained by the camera positioned in the final position of the path), the same point will be located at ^Nf_i position (this notation is illustrated in Figure 4).

Matrix $^k\mathbf{R}$ indicates the rotation between an intermediate position in the path and the final one, whereas ^kt_d is the scaled translation of the camera between the two positions:

$$^kt_d = \frac{^kt}{^N\xi} \tag{9}$$

where $^N\xi$ is the distance from Π to the origin of the camera placed at the target position.

The projective homography $^k\mathbf{G}$ is defined as:

$$^k\mu_i \, ^kf_i = \, ^k\mathbf{G}\, ^Nf_i + \beta_i \, ^ke \tag{10}$$

where $^k\mu$ is a scale factor, $^ke = \mathbf{A}\, ^k\mathbf{R}\, ^kt_d$ is the projection in the image captured by the camera at time k of the optical centre when the camera is in the target position, and β_i is a constant scale factor which depends on the distance $^N\xi$:

$$\beta_i = \frac{d(P_i, \Pi)}{^N\xi} \tag{11}$$

where $d(P_i, \Pi)$ is the distance from the contact surface plane to the 3D point.

Although β_i is unknown, applying (10) between the initial camera position (the one obtained when the collision is detected indicated from now on by the superscript 0) and the target camera position (indicated by the superscript N) we can obtain that:

$$\beta_i = sign\left(\frac{\left(^0\mu_i \, ^0f_i - \, ^0\mathbf{G}\, ^Nf_i\right)_1}{\left(\mathbf{A}\, ^0\mathbf{R}\, ^0t_d\right)_1}\right)\frac{\left\|^0\mathbf{G}\, ^Nf_i \wedge \, ^0f_i\right\|}{\left\|\mathbf{A}\, ^0\mathbf{R}\, ^0t_d \wedge \, ^0f_i\right\|} \tag{12}$$

where subscript 1 indicates the first element of the vector and $\mathbf{A}$ is a matrix having the estimated camera internal parameters, as it was defined in (1) in Section 2.

Replacing (12) in (10) we can obtain the image coordinates of the features just by dividing ${}^{k}\mu_i {}^{k}\boldsymbol{f}_i$ by its last component (this way we obtain the homogeneous coordinates of ${}^{k}\boldsymbol{f}_i$). Thus, at time k we can compute the coordinates in pixels of a feature through the Equation:

$$
{}^{k}\mu_i {}^{k}\boldsymbol{f}_i = {}^{k}\mathbf{G}^{N}\boldsymbol{f}_i + sign\left(\frac{\left({}^{0}\mu_i {}^{0}\boldsymbol{f}_i - {}^{0}\mathbf{G}^{N}\boldsymbol{f}_i \right)_1}{\left(\mathbf{A}\,{}^{0}\mathbf{R}\,{}^{0}t_d \right)_1} \right) \frac{\left\| {}^{0}\mathbf{G}^{N}\boldsymbol{f}_i \wedge {}^{0}\boldsymbol{f}_i \right\|}{\left\| \mathbf{A}\,{}^{0}\mathbf{R}\,{}^{0}t_d \wedge {}^{0}\boldsymbol{f}_i \right\|} \mathbf{A}\,{}^{k}\mathbf{R}\,{}^{k}t_d \tag{13}
$$

As it can be seen in equation (13), in order to compute ${}^{k}\boldsymbol{f}_i$ we must obtain homography matrices ${}^{0}\mathbf{G}$ and ${}^{k}\mathbf{G}$, as well as rotation ${}^{0}\mathbf{R}$ and scaled translation ${}^{0}t_d$.

If P_i is on the contact surface, β_i is null. Therefore, applying (10) between initial and final positions in the path we have:

$$
{}^{0}\mu_i {}^{0}\boldsymbol{f}_i = {}^{0}\mathbf{G}^{N}\boldsymbol{f}_i \tag{14}
$$

Projective homography relating initial and final positions ${}^{0}\mathbf{G}$ can be obtained through expression (14) if at least four points on the surface contact plane are given [15]. In order to obtain ${}^{0}\mathbf{R}$ and ${}^{0}t_d$ we must introduce the concept of the Euclidean homography matrix $\mathbf{H}$.

From projective homography $\mathbf{G}$ it can be obtained the Euclidean homography $\mathbf{H}$ as follows:

$$
\mathbf{H} = \mathbf{A}^{-1}\mathbf{G}\mathbf{A} \tag{15}
$$

From $\mathbf{H}$ it is possible to determine the camera motion applying the algorithm shown in [16]. So, applying this algorithm between initial and target positions we can compute ${}^{0}\mathbf{R}$ and ${}^{0}t_d$.

This way we have obtained ${}^{0}\mathbf{G}$, ${}^{0}\mathbf{R}$ and ${}^{0}t_d$. Now, it is necessary to compute ${}^{k}\mathbf{G}$ in order to obtain the position of the features in the image plane according to the planned path. To do

so, Euclidean homography matrix $^k\mathbf{H}$ must be previously obtained. Then, it is easy to compute $^k\mathbf{G}$ from Equation (15):

$$^k\mathbf{G} = \mathbf{A}\,^k\mathbf{H}\mathbf{A}^{-1} \tag{16}$$

$\mathbf{H}$ can be decomposed into a rotation matrix and a rank 1 matrix:

$$\mathbf{H} = \mathbf{R} - \mathbf{R}t_{\mathrm{d}}n \tag{17}$$

Applying Equation (17) we can obtain $^k\mathbf{H} = {}^k\mathbf{R} - {}^k\mathbf{R}\,^kt_{\mathrm{d}}n$ (at any position of the planned path). Where $^k\mathbf{R}$ and $^kt_{\mathrm{d}}$ are obtained by the path planning [3] as we have previously described. The different steps described in this section to obtain the trajectory to be followed are shown in Figure 5.
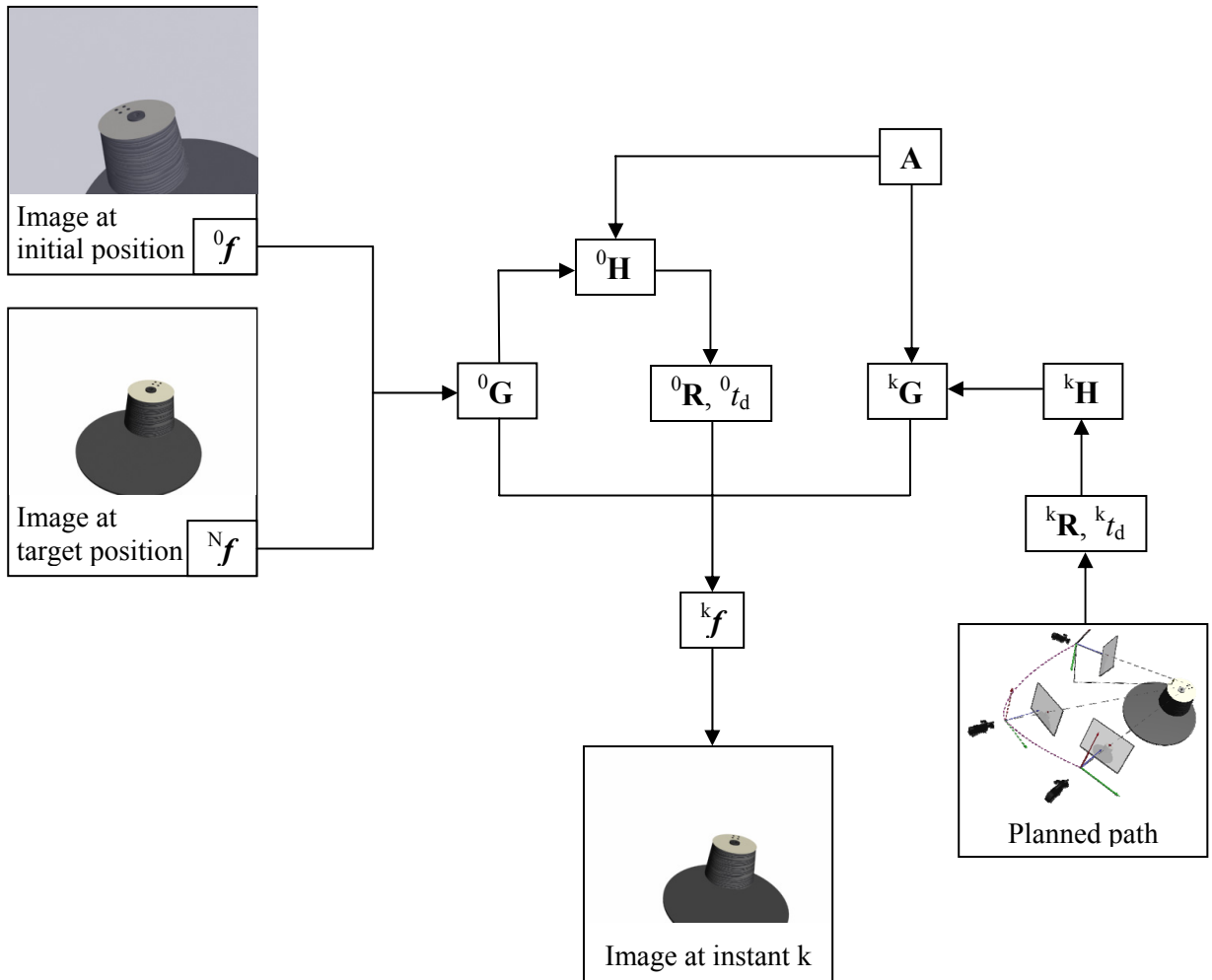
Figure 5. Block diagram of the path modification after the robot interaction with the environment.

This method is probed to be independent from errors in estimation of intrinsic parameters. This occurs only under assumption that the initial error on the estimated Euclidean homography is propagated along the trajectory.

We can compute the estimated initial Euclidean homography matrix $^0\hat{\mathbf{H}}$ applying equation (15) in this way:

$$^0\hat{\mathbf{H}} = \hat{\mathbf{A}}^{-1} \, {}^0\mathbf{G}\hat{\mathbf{A}} \tag{18}$$

Substituting the value of $^0\mathbf{G}$ obtained from equation (16) the estimated initial Euclidean homography matrix $^0\hat{\mathbf{H}}$ can be computed from the estimated camera intrinsic parameters $\hat{\mathbf{A}}$, the real camera intrinsic parameters $\mathbf{A}$ and the initial real Euclidean homography $\mathbf{H}$:

$$^0\hat{\mathbf{H}} = \hat{\mathbf{A}}^{-1}\mathbf{A} \, {}^0\mathbf{H}\mathbf{A}^{-1}\hat{\mathbf{A}} \tag{19}$$

The assumption previously related indicating that the initial error on the estimated Euclidean homography is propagated along the trajectory can be expressed in a given iteration k as:

$$^k\hat{\mathbf{H}} = \mathbf{E}_A \, {}^k\mathbf{H}\mathbf{E}_A^{-1} \tag{20}$$

where $\mathbf{E}_A = \hat{\mathbf{A}}^{-1}\mathbf{A}$.

According to (13) the estimated pixel feature coordinates are:

$$^k\hat{\mu}_i \, {}^k\hat{\boldsymbol{f}}_i = \hat{\mathbf{A}} \, {}^k\hat{\mathbf{H}}\hat{\mathbf{A}}^{-1} \, {}^N\boldsymbol{f}_i + sign\left(\frac{\left({}^0\mu_i \, {}^0\boldsymbol{f}_i - {}^0\mathbf{G} \, {}^N\boldsymbol{f}_i \right)_1}{\left(\hat{\mathbf{A}} \, {}^0\hat{T} \right)_1} \right) \frac{\left\| {}^0\mathbf{G} \, {}^N\boldsymbol{f}_i \wedge {}^0\boldsymbol{f}_i \right\|}{\left\| \hat{\mathbf{A}} \, {}^0\hat{T} \wedge {}^0\boldsymbol{f}_i \right\|} \hat{\mathbf{A}} \, {}^k\hat{T} \tag{21}$$

where the estimated vector $^k\hat{T}$ is:

$$\hat{^kT} = \left\| n\left(\hat{\mathbf{A}}^{-1}\mathbf{A}\right)^{-1} \right\| \hat{\mathbf{A}}^{-1}\mathbf{A}^{\,k}\mathbf{R}^{\mathrm{T}\,k}t_{\mathrm{d}} \tag{22}$$

As it is demonstrated in [2] the second term of the sum in Equation (21) is equal to the second term of the sum in (13). Applying Equation (20) in the first term of the sum in (21) it is easy to obtain that:

$$\hat{\mathbf{A}}^{\,k}\hat{\mathbf{H}}\hat{\mathbf{A}}^{-1\,\mathrm{N}}\boldsymbol{f}_{\mathrm{i}} = \hat{\mathbf{A}}\mathbf{E}_{\mathrm{A}}^{\;\;k}\mathbf{H}\mathbf{E}_{\mathrm{A}}^{-1}\hat{\mathbf{A}}^{-1\,\mathrm{N}}\boldsymbol{f}_{\mathrm{i}} = \mathbf{A}^{\,k}\mathbf{H}\mathbf{A}^{-1\,\mathrm{N}}\boldsymbol{f}_{\mathrm{i}} \tag{23}$$

Considering (16) we obtain $^k\mathbf{G}\,\boldsymbol{f}_{\mathrm{i}}$ which is just the first term of the sum in (13). This way it is demonstrated that:

$$^k\hat{\mu}_{\mathrm{i}}^{\;\;k}\hat{\boldsymbol{f}}_{\mathrm{i}} = {}^k\mu_{\mathrm{i}}^{\;\;k}\boldsymbol{f}_{\mathrm{i}} \tag{24}$$

Therefore, the method is not affected by errors in intrinsic parameters.

6   RESULTS

In this section the results obtained from the application of the proposed system to an insertion task is described. It requires the joint use of the information obtained from the force sensor (control of the robot's interaction with its workspace) together with that obtained from the computer vision system (tracking of the desired trajectory during the insertion, using movement flow-based visual servoing). Firstly, we describe the experimental setup. Subsequently, the results obtained from using movement flow-based visual servoing in the tracking of different trajectories are shown. The system robustness with respect to camera calibration errors is demonstrated in these experiments. Finally, different insertion experiments that require the fusion of visual and force information are described. These experiments require to modify the image trajectory from the force information as described in Section 5.

16

6.1 Experimental setup

In Figure 6.a, the robotic system employed is shown.
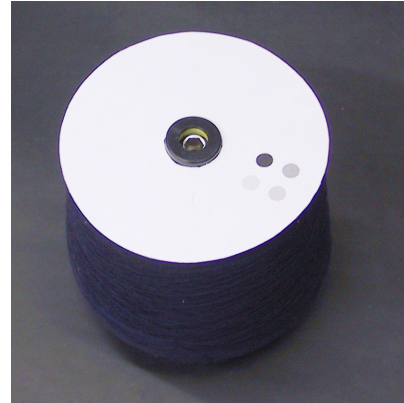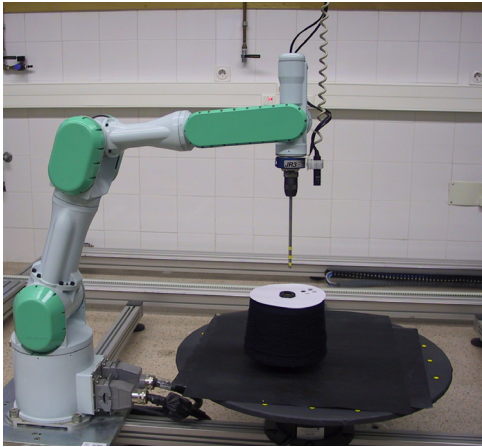
a)   b)

Figure 6. Experimental setup.

The characteristics of each of the devices are:

- "Eye-in-hand" system: The capturing of images from the robot end-effector is carried out by a JAI-M536 mini-camera with a remote optical head. MATROX GENESIS is used as image acquisition and processing board. The camera is able to acquire up to 30 frames/second and is previously submitted to a calibration process (focal length is 7,5 mm).

- Manipulation system: This is composed of a 7 d.o.f. Mitsubishi PA-10 robot equipped with a force sensor (67M25A-I40 from JR3. Inc.). The robot has a bar at the end-effector.

- Target: Since we are not interested in image processing issues in this paper the tracked target is composed by four grey marks. They will be the extracted features during the tracking. As it can be seen in Figure 6.b, the orifice for the insertion has two different diameters.

6.2 Experimental results

We describe the results obtained from the tracking of trajectories and show the behaviour of the system at the moment when the insertion is carried out, in other words, by combining the information from the force sensor with that of the computer vision system.

6.2.1 Tracking trajectories

This first experiment consists of the tracking of a parabolic trajectory between the points (327, -330, -82) and (0, -130, 0), expressed in millimetres. Figure 7 shows a sampling of the desired Cartesian trajectory followed by the camera in the 3D space. It also shows the corresponding sampling of the desired trajectory in the image $^{k}\mathbf{s}_{d}$ (Section 2 describes how the computation of this desired trajectory is done). Figure 7.a shows the evolution of the coordinate frame at the robot end-effector during the desired trajectory.

a)   b)
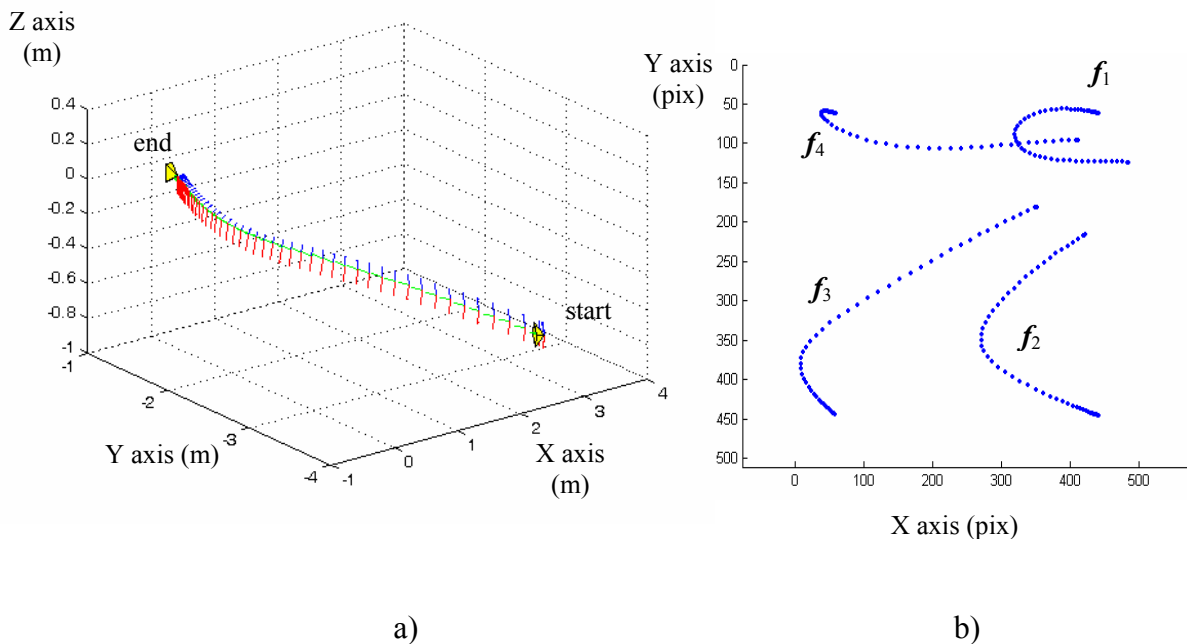
Figure 7. Sampling of the desired trajectory in the 3D Cartesian space and in the image.

Obviously, as it is shown in Figure 8, the correct tracking of the previous mentioned desired trajectory is not possible using a classical image-based visual servoing system (considering the desired features the ones observed in the final position). Using a classical image-based

18

visual servoing the features tend to follow a straight line, not allowing the correct tracking.
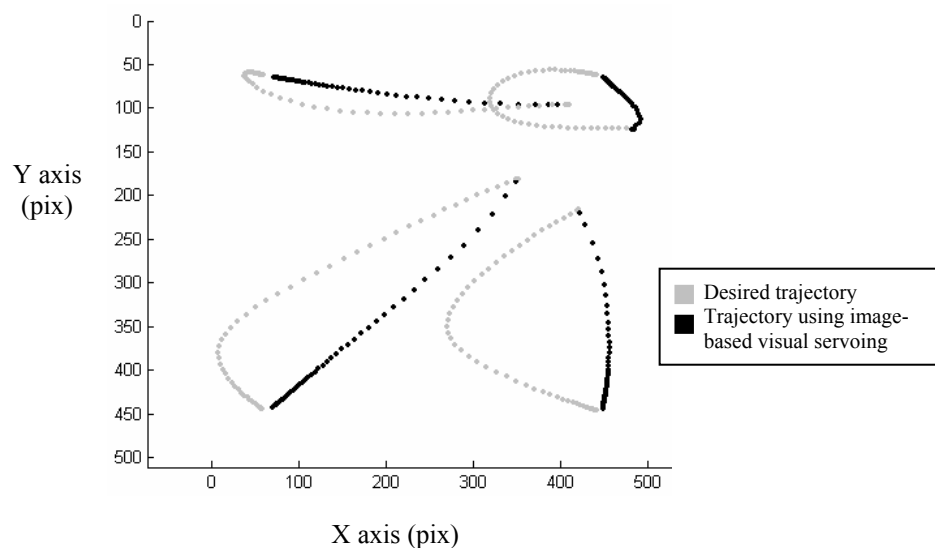

Figure 8. Sampling of the desired trajectory in the image and the obtained using image-based visual servoing.

Now, we consider that the movement flow-based visual servoing approach is applied (In [3] more details about the different parameters of this method are described). In Figure 2.a a detail of the movement flow for the feature f_1 is illustrated as an example of the movement flow obtained. Figure 9.a shows the desired trajectory and the one obtained in the 3D Cartesian space. To evaluate the tracking, Figure 9.b shows the mean error measured in pixels of the features in the image. That is, $E_m = \dfrac{1}{M} \sum_{i=1}^{M} \sqrt{E_{xi}^2 + E_{yi}^2}$. From the graph in Figure 9.a, it can be concluded that the system carries out correctly the tracking in the 3D space (the error remains small during the tracking).

.
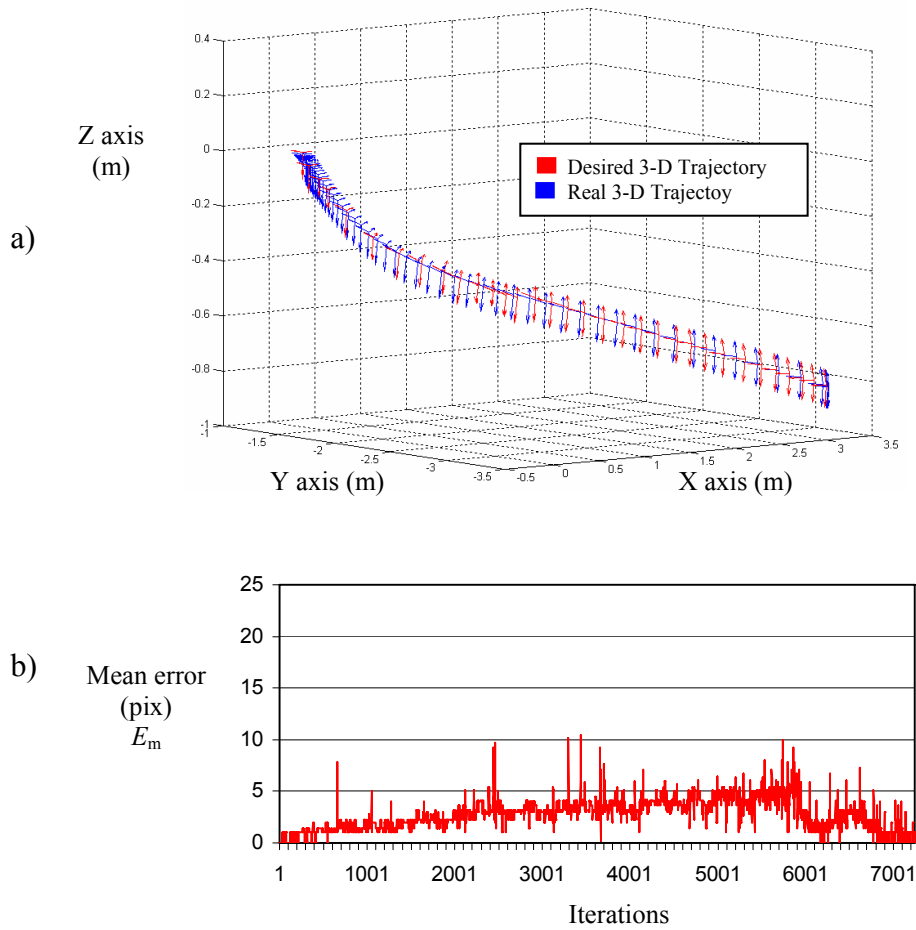
Figure 9. a) Comparison of the real trajectory and the sampling of the desired trajectory in the 3D space. b) Average error in the image.

It is well known that image-based visual servoing is locally stable. Therefore, it only ensures the correct convergence if the desired configuration is sufficiently near the current one. As it is shown in Figure 9.b, error remains small, thus assuring that local minimum does not exist in the tracking.

6.2.2 Tracking trajectories with interaction. Robustness with respect to errors in intrinsic parameters.

This section shows a new experiment in which the desired image trajectory is modified depending on the interaction forces (see Section 5). Now, an interaction appears during the tracking of the trajectory described in the previous section. This interaction corresponds with

an obstacle which obstructs the trajectory during a certain time. Due to the interaction forces the desired trajectory in the image is modified. The new desired trajectory is shown in Figure 10. Figure 11 shows the tracking of the desired trajectory shown in Figure 10 (to develop the tracking movement flow-based visual servoing is used). In order to demonstrate the system robustness with respect to errors in the employed intrinsic parameters an error of 40% is introduced on the intrinsic parameters. In this experiment, we can observe that the system carries out correctly the tracking of the trajectory independently of the non-accuracy determination of the intrinsic parameters.
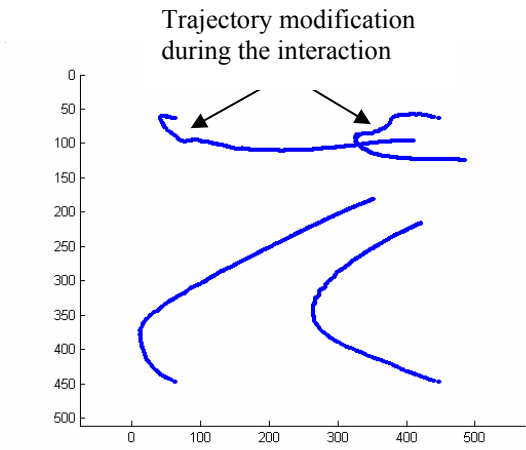
Figure 10. Modified desired trajectory.
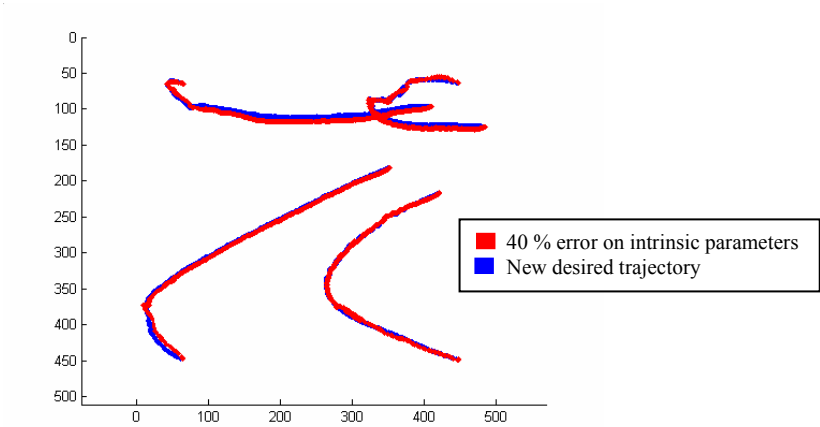
Figure 11. Tracking with 40% of error in intrinsic parameters.

6.2.3 Fusing visual and force information

Because of the high precision required for the insertion, a perfect insertion can not be guaranteed by considering only visual information. As it is shown in Figure 12, using the visual-force control strategy proposed in our previous works [3] the insertion can not be developed correctly. The slightest contact with the edge of the aperture causes the forces to rapidly increase and the insertion is not carried out. Therefore, this previous approach is not adequate when new contact points appear during the task. To solve this problem, in the paper a new approach based on impedance control to combine visual and force information is proposed. To demonstrate the correct behaviour of the system using simultaneously force and visual information, two experiments, in which the insertion is carried out in different positions, will be described. In Figure 13, the trajectory of the bar at the robot end-effector is represented during both insertions. The aperture in which the insertion is done is shown to verify the correct behaviour.
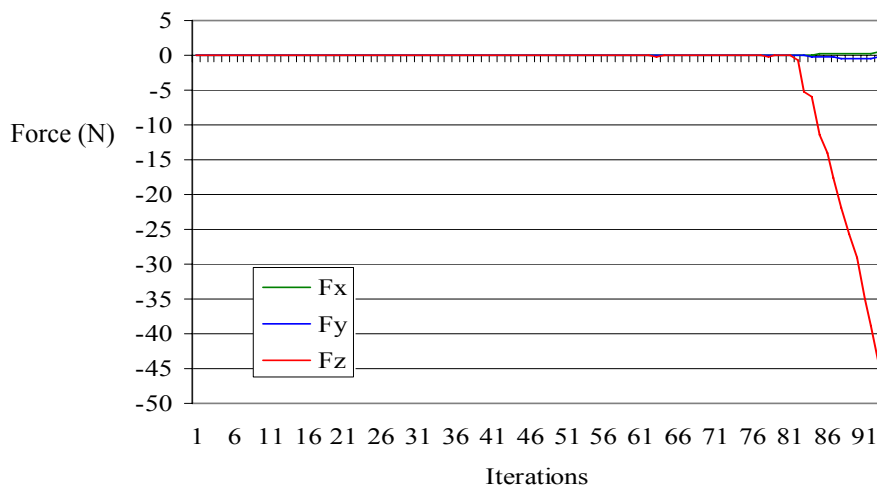

Figure 12. Forces obtained using the visual-force control proposed in [3].
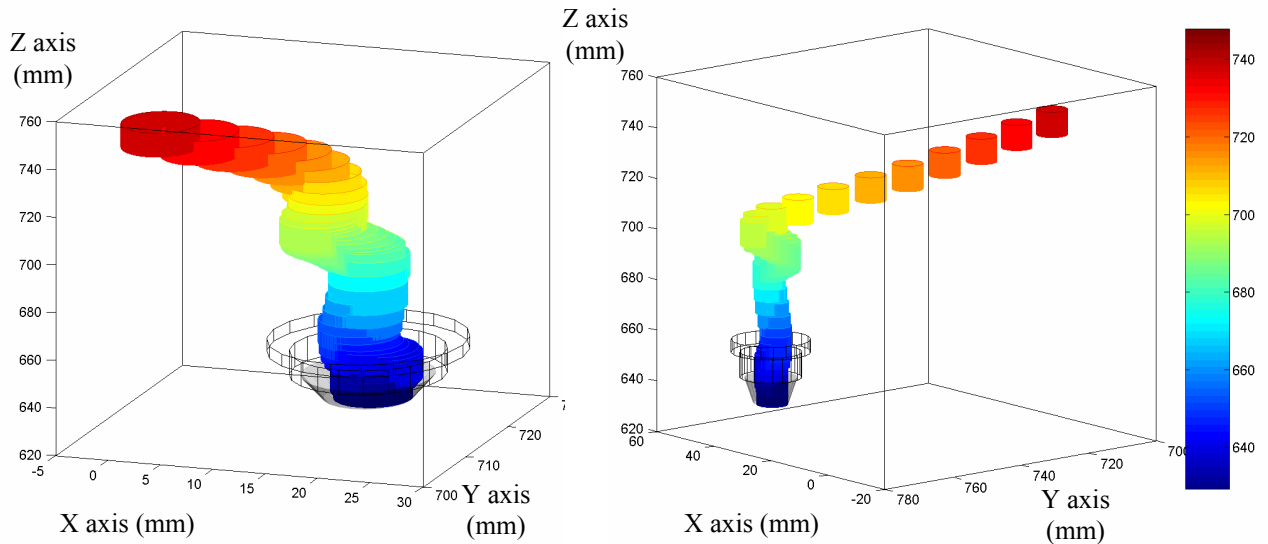
Figure 13. Trajectory followed by the robot in two different insertion experiments.

In Figure 14, the rapid correction of the defects in the trajectory generated by the movement flow-based visual servoing system is observed. In fusing the force and the visual information, the forces are kept low throughout the trajectory. The forces are rapidly compensated for, and the trajectory is accurately corrected, thus permitting the insertion. One important aspect which guarantees the correct behaviour in visual-force applications is the non-time dependent behaviour (see [3] for more details).
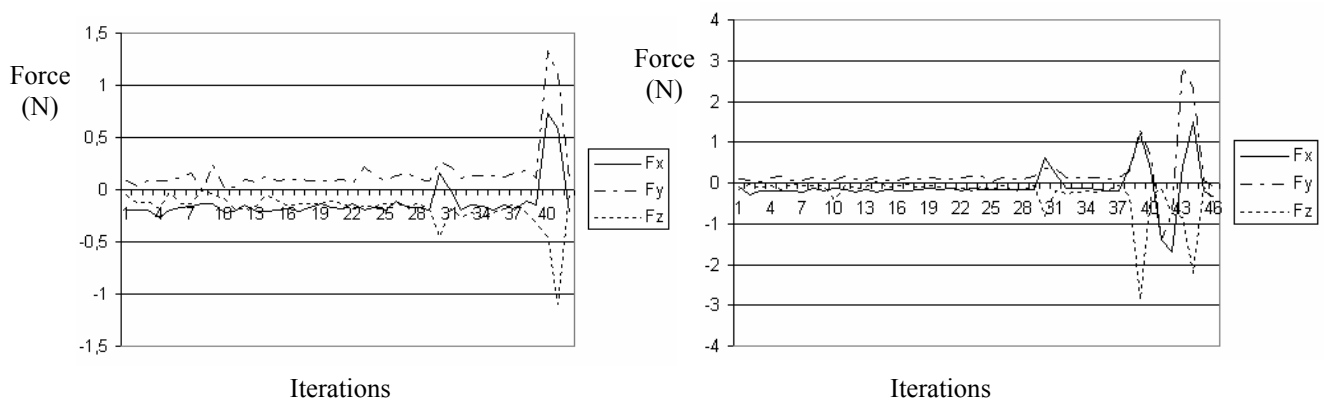

Figure 14. Evolution of the interaction force of the robot with its workspace in the two experiments.

23

If the trajectory is not well generated, the insertion cannot be correctly developed. Therefore, the image trajectory must be modified depending on the interaction forces. This aspect is illustrated in Figure 15. In this case, the robot collides with the surface. Once this surface is recognized, the image trajectory is modified (see Section 5) allowing the joint use of both sensors to control the task.
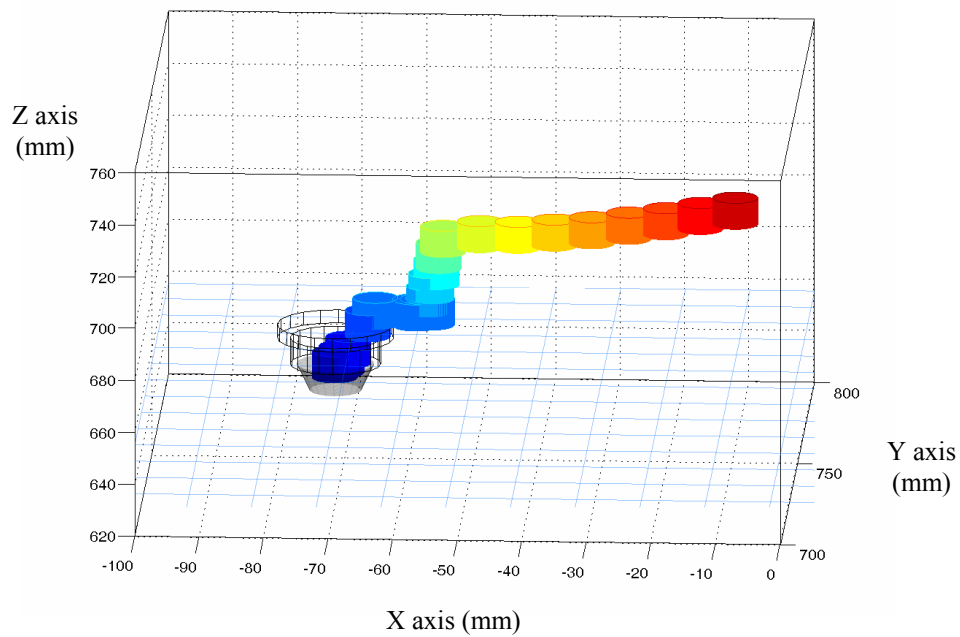

Figure 15. 3D Cartesian trajectory during an insertion task when the robot collides with the surface.

The 3D Cartesian trajectory that the robot must follow during the insertion task represented in Figure 15 can be seen in the image space in Figure 16. This trajectory needs to be modified by the fact that the robot collides with the surface of the object before it complete the insertion task. As Section 5 describes, once the system detects the collision the image trajectory is re-planned so that the insertion can be completed following the nearest trajectory allowed by the surface restrictions. Doing so, the new desired image trajectory obtained is shown in Figure 17. Applying movement flow-based visual servoing to track this new desired trajectory, the 3D trajectory shown in Figure 15 is obtained.
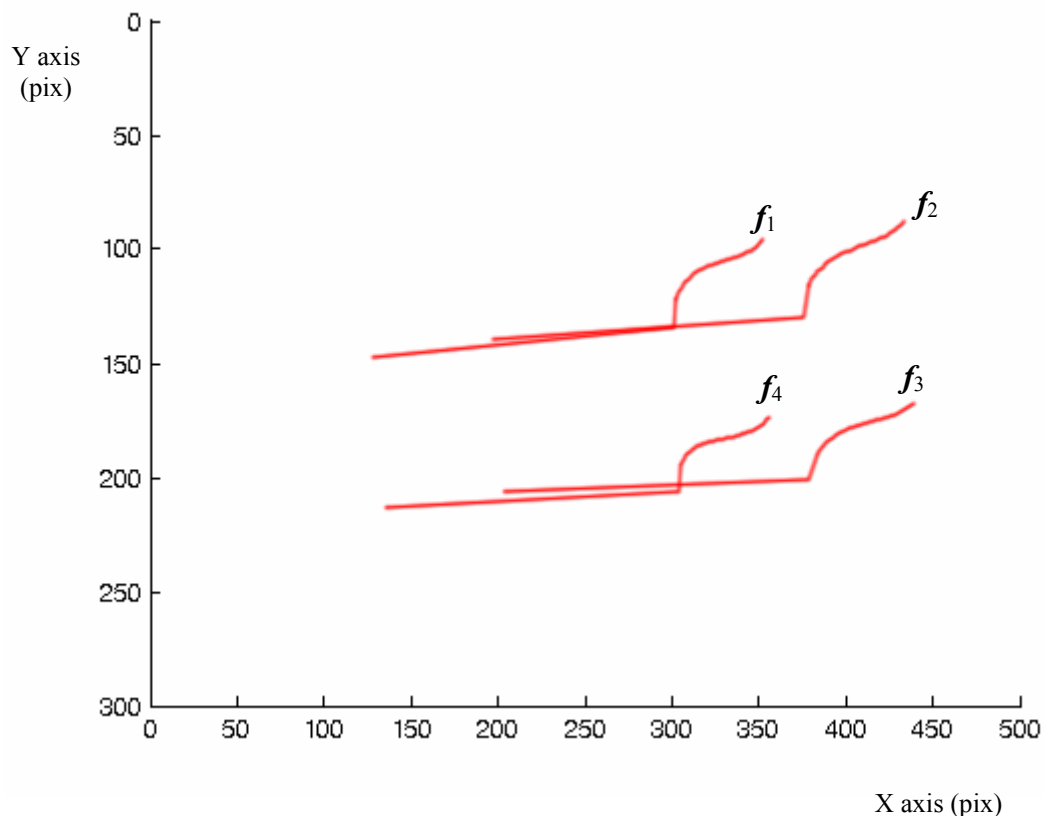
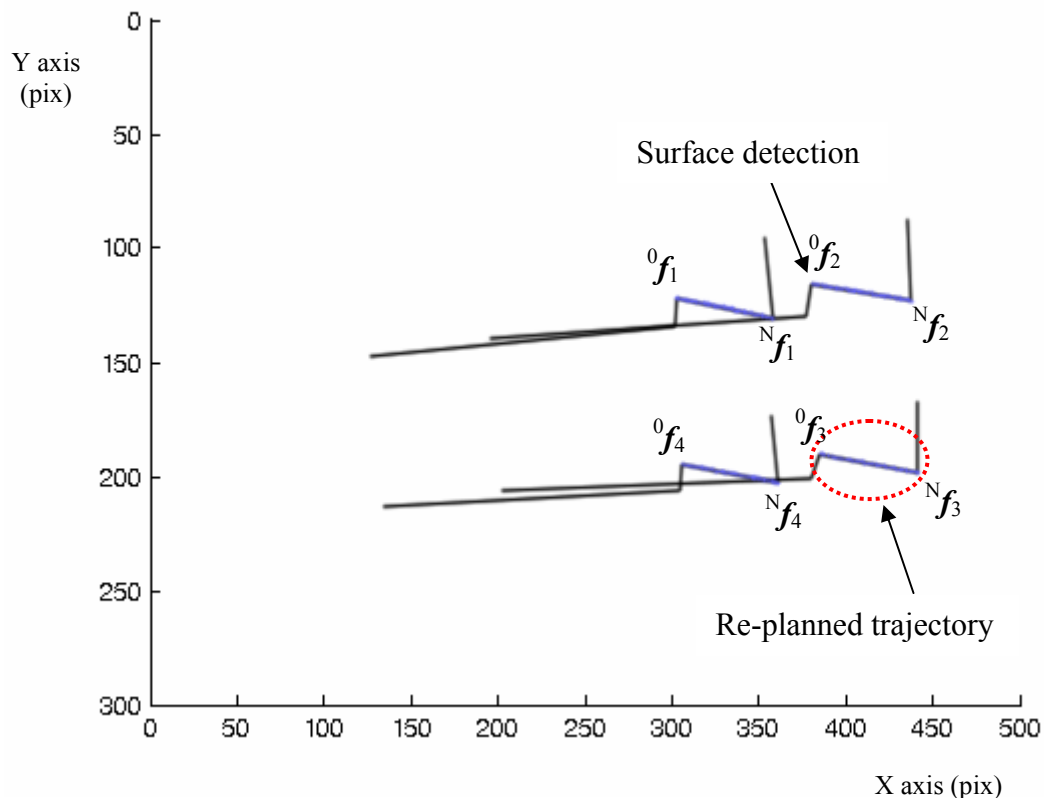Figure 16. Desired image trajectory during an insertion task.

Figure 17. Trajectory followed by the robot during an insertion task when it collides with the surface.

7   CONCLUSIONS

A new method to fuse sensorial information from a computer vision system with that from a force sensor has been described. This method does not require previous geometric knowledge about the workspace to develop the task. The method was applied to an insertion process that requires great precision. In this application, not only it is necessary to obtain given features in the image from the initial ones, but also a tracking of the desired trajectory between them (i.e. fulfilling the desired spatial restrictions) is necessary for the correct insertion to be carried out.

A technique to track trajectories, called movement flow-based visual servoing has been employed. This method avoids the use of information about time to ensure the correct tracking. An important aspect of the proposed visual-force control system is the possibility to carry out the task when the camera is not correctly calibrated. In the results section the systems have been probed to be robust when errors in the camera intrinsic parameters exist.

The paper demonstrates the correct behaviour under interaction between the robot end-effector and the workspace. The planned trajectory in the image space has been modified in order to allow the system to achieve the desired position when a collision is detected. This new trajectory in the image space is computed without the necessity of knowing the correct camera calibration parameters.

REFERENCES

1. Hutchinson, S., Hager, G. D. and Corke, P. I.: A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, **12(5)**, 651-670 (1996).

2. Mezouar, Y. and Chaumette, F.: Path Planning For Robust Image-based Control, IEEE Transactions on Robotics and Automation, **18(4)**, 534-549 (2002).

3. Pomares, J., Torres F.: Movement-flow based visual servoing and force control fusion for manipulation tasks in unstructured environments. IEEE Transactions on Systems, Man, and Cybernetics—Part C. **35(1)** 4 – 15 (2005).

4. Nandi, G. and Mitra, D.: Fusion Strategies for Minimizing Sensing-Level Uncertainty in Manipulator Control. Journal of Intelligent and Robotic Systems, **43(1)**, 1-32 (2005).

5. Baeten, J. and Schutter, J. De.: Hybrid Vision/Force Control at Corners in Planar Robotic-Contour Following. IEEE/ASME Transactions on Mechatronics. **7(2)**, 143 - 151 (2002).

6. Nelson, B. J. and Khosla, P. K.: Force and vision resolvability for assimilating disparate sensory feedback, IEEE Transaction on Robotics and Automation. **12(5)**, 714-731 (1996).

7. Zhou, Y., Nelson, B. J. and Vidramaditya, B.: Fusing force and vision feedback for micromanipulation. In: Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, pp. 1220-1225 (1998).

8. Collani, Y. von, Scheering, C., Zhang, J. and Knoll. A.: A neuro-fuzzy solution for integrated visual and force control. In: Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, MI, pp. 2965-2970 (1999).

9. Morel, G., Malis, E. and Boudet. S.: Impedance based combination of visual and force control. In: Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, pp. 1743-1748 (1998).

10. Pomares, J., Torres, F. and Puente, S. T.: Disassembly movements for geometrical objects through heuristic methods. In: Proceedings of the SPIE International Symposium on

Intelligent Systems and Advanced Manufacturing, Boston, MA, vol. 4569, pp. 71-80 (2002).

11. Pomares, J., Puente, S. T., Torres, F., Candelas, F. A., Gil, P.: Virtual disassembly of products based on geometric models, Computers in industry. **55(1)**, 1-14 (2004).

12. Nelson, B. J., Morrow, J. and Khosla, P. K.: Robotic manipulation using high bandwith force and vision feedback. Mathl. Computer Modelling. **24(5/6)**, 11-29 (1996).

13. Whitney, D. E.: Force feedback control of manipulator fine motions. Journal of Dynamic Systems, Measurement and Control. 97-97 (1977).

14. Espiau, B., Merlet, J.P. and Samson, C.: Force-feedbackcontrol and non-contact sensing: a unified approach. In: Proceedings of the 8th CISM-IFTOMM Symposium on Theory and Practice of Robots Manipulators, (1990).

15. Hartley R., Zisserman A.: Multiple view Geometry in Computer Vision. Cambridge University Press, 91-92 (2000).

16. Zhang, Z.: Three-dimensional reconstruction under varying constraints on camera geometry for robotic navigation scenarios, Ph.D. thesis. 108-140 (1996).