Movement-Flow-Based Visual Servoing and Force Control Fusion for Manipulation Tasks in Unstructured Environments

Jorge Pomares and Fernando Torres, Member, IEEE

Abstract—In this paper, a new approach for fusing visual and force information is shown. First, a new method for tracking trajectories, called movement flow-based visual servoing system, which presents the correct behavior in the image and in the three-dimensional space, is described. The information obtained from this system is fused with that obtained from a force control system in unstructured environments. To do so, a new method of recognizing the contact surface and a system for fusing visual and force information are described. The latter method employs variable weights for each sensor system, depending on a criteria based on the detection of changes in the interaction forces processed by a Kalman filter.

Index Terms—Autonomous manipulation, eye-in-hand system, force control, multisensor systems, tracking trajectories, 2-D visual servoing.

I. INTRODUCTION

This paper presents a new approach for fusing visual and force information and its application to manipulation tasks in unstructured environments. Using a classical image-based visual servoing system, a three-dimensional (3-D) trajectory between the initial and the desired configurations cannot be specified (especially for certain cases of large rotational differences [5]) and most of the applications in which it is generally used are point-to-point based [6]. Only the desired configuration is indicated and not the trajectory that should be followed to arrive at such a configuration. This reason motivates that to control the position of the robot during the task, a new approach, called movement flow-based visual servoing has been developed, which allows the tracking of the desired trajectory between the initial and the desired configurations. This system is employed to track trajectories previously generated in the image space and also has the correct sort of behavior in the 3-D space. This approach uses what we call a movement flow to determine the desired configuration from the current one. Only recently [17] has it been possible to find visual servoing applications for tracking trajectories in the image. However, in such approaches, the tracking is formulated as a timed trajectory in the image and, therefore, the current configuration and the desired one are separated by a time interval $\Delta t$. As such, if an image-based control system is employed to track timed trajectories, the system risks not following the desired trajectory at the cost of trying to maintain time restrictions (see Section V-B). To resolve this problem, the so-called movement flow-based visual servoing is used, with which the task to be carried out by the robot is encoded in the image space. We should also mention the use of virtual fixtures [9] to guide the robot. This method, however, is not used to track a given trajectory in the image, but rather to guide the robot toward a point, a line, or a surface, introducing vision-based motion constraints. The use of virtual fixtures generates a set of preferred directions to achieve a given configuration, avoiding the geometric constraints imposed by sensor data.

Due to the different nature of the magnitude measured with the visual and force systems, a great number of approaches employed for fusing the information obtained from both sensors have been based, up to now, on hybrid control [24]. Concerning hybrid visual-force systems, we should mention studies like [1] and [2], which extend the “task frame” formalism [4], [8]. Within this field, we should also mention [12], which is applied for the tracking of unknown surfaces. In [28], a similar strategy for the tracking of surfaces in unstructured environments is shown. In [19], a system for grasping objects in real time, which employs information from an external camera and that obtained from the force sensors of a robotic hand, is described. Another strategy used for the combination of both sensory systems is the use of impedance control. Thus, based on the basic scheme of impedance control, we should mention several modifications like the one described in [18], which adds an external control loop that consists of a visual controller which generates the references for an impedance control system. We can mention other approaches like the one described in [20] in which the force and visual sensors act at the same level of the control hierarchy. In works such as [26], the use of virtual forces applied to approaching tasks without contact, is proposed. In [16], the combination of different sensor systems, employing the task function approach, is described. We should also mention other techniques, such as the concept of “resolvability” [20], which affords a measurement of each sensors’ ability to resolve the movement.

Up to now, the approaches for fusing visual and force information do not consider the possibility of both sensors providing contradictory information at a given moment of the task. Thus, in unstructured environments, the visual servoing system can establish a movement direction that is impossible according to the interaction information obtained from the force sensor. In this...
paper, we consider this possibility and the sensory information obtained is processed to allow the use of both sensors for controlling the robot. Furthermore, in this article, movement flow is employed to combine the two different types of sensor data. This aspect allows us to guarantee that the 3-D trajectory imposed by the visual servoing system will be coherent with the spatial restrictions.

This paper is organized as follows: The main characteristics of the trajectory to be tracked and the notation used is first described in Section II. Section III shows the main concepts concerning the generation and use of the movement flow obtained from a trajectory in the image. In Section IV, the strategy used for the fusion of force information with that from the movement flow-based visual servoing is described. In Section V, experimental results, using an eye-in-hand camera system, confirm the validity of the proposed algorithms. The final section presents the main conclusions.

II. TRAJECTORY TO BE TRACKED AND NOTATION

In this paper, the generation of the trajectories specified in the image space are not described, so that the presence of a planner, which provides the robot with the 3-D trajectory $\gamma(t)$ to be tracked (i.e., the desired 3-D trajectory of the camera at the end-effector), is assumed. For this study, we have employed planners already described in our previous studies [22], [23]. These trajectories are generated from a 3-D geometric model of the workspace, so that it is necessary to design a visual servoing system that performs the tracking of the 3-D trajectory using visual information and, at the same time, tests whether it is possible to carry out such tracking, depending on the interaction forces obtained. The use of strategies like the one described in [9], based on virtual fixtures, allows the guidance of the robot, avoiding the obstacles in the workspace, but it does not allow the tracking of the trajectory previously established.

By sampling $\gamma(t)$, a sequence of $N$ discrete values is obtained, each of which represents $N$ intermediate positions of the camera $k_{\gamma}/k \in 1 \ldots N$. From this sequence, the discrete trajectory of the object in the image $S = \{k_{\mathbf{s}}/k \in 1 \ldots N\}$ can be obtained, where $k_{\mathbf{s}}$ is the set of $M$ points or features observed by the camera at instant $k$. $k_{\mathbf{s}} = \{k_{\mathbf{f}}/i \in 1 \ldots M\}$.

The discrete trajectory in the image that must be tracked by the movement flow-based visual servoing system is composed of $M$ trajectories $C_{i}/i \in 1 \ldots M$ (one for each feature extracted in the image), so that the values in the image of each of the $M$ trajectories at the instants $k$ are $C_{i} = \{k_{\mathbf{f}}/k \in 1 \ldots N\}$.

The desired trajectories of a robot in the image are often specified using timed trajectories. In this case, the objective of the control is to carry out the tracking using a time variable reference. However, in applications in which the robot must interact with the workspace, the trajectory can be obstructed. Therefore, the references can be delayed not allowing the correct tracking (see Section V-B). Thus, for each feature, a codification of the desired trajectory in the image $f_{\mathbf{id}}(\tau)$ is done, omitting the information about time, so that $f_{\mathbf{id}} : \Gamma' \rightarrow \mathbb{S}$ where $\Gamma' \subset \mathbb{R}$. For this reason, to obtain the desired trajectory $f_{\mathbf{id}}(\tau)$, a cubic cardinal spline interpolator is applied to the samples of $C_{i}$, and a representation of this trajectory in the image space is obtained [Fig. 1(a)].

The following notations are used. The commanded velocity for the movement flow-based visual servoing and for the force control systems are $\mathbf{v}_{\mathbf{id}}$ and $\mathbf{v}_{\mathbf{id}}^F$, respectively ($\mathbf{v}$ are the angular velocities). $F(f_{x}, f_{y}, f_{z}, n_{x}, n_{y}, n_{z})$ are force (N) and torque (N m) exerted by the environment onto the robot and $k$ is the tool stiffness (N m or N m rad$^{-1}$). $\lambda_{M}$ and $\lambda_{F}$ are the proportional control gains for the visual and force controllers, respectively. $\mathbf{s} = [f_{x1}, f_{x2}, \ldots, f_{xM}]^T$ are the features extracted from the image and, for a feature $i$, $f_{xi}$, the coordinates of the nearest point to it in the desired trajectory are $f_{\mathbf{id}} = (f_{xid}, f_{yid})$. Finally, $\Phi_{i}$ is the movement flow for the feature $i$ as described in Section III.

III. MOVEMENT FLOW

In this section, the concept of movement flow is described, showing the appropriate considerations for its implementation.
To do so, the general concept of movement flow is shown in Section III-A, and then the potential and the weight functions employed are described in Sections III-B and III-C, respectively.

### A. Definition

The movement flow $\Phi$ is defined as $\Phi : \mathcal{M} \rightarrow \mathcal{T}$, where $\mathcal{M}$ is the image space and $\mathcal{T}$ is the image tangent space. In general, the movement flow has the following properties: its values at each point of the desired image trajectory are tangent to it and those outside the trajectory aim to decrease the tracking error. Therefore, the 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, permitting the tracking of the trajectory. Thus, considering image-based control, the velocity applied to the robot with respect to the camera coordinate frame will be

$$\mathbf{v}_M^i = -\lambda_M \cdot \mathbf{J}_i^+ \cdot \mathbf{e}_i$$  \hspace{1cm} (1)

where $\lambda_M$ is the gain of the proportional controller; $\mathbf{J}_i^+$ is the pseudoinverse of the interaction matrix $[13]$, $\mathbf{e}_i = \mathbf{s} - \mathbf{s}_i$, $\mathbf{s} = \begin{bmatrix} f_1, f_2, \ldots, f_M \end{bmatrix}^T$ are the features extracted from the image $s_x^i$ $= \begin{bmatrix} f_1 + m_1 \Phi_1(f_1), f_2 + m_2 \Phi_2(f_2), \ldots, f_M + m_M \Phi_M(f_M) \end{bmatrix}^T$.

$\Phi_i$ is the movement flow for the feature $i$, and $m = \{m_1, m_2, \ldots, m_M\}$ determines the progression speed. As can be seen, the error is measured directly in the image, which implies a certain degree of robustness with respect to modeling errors and noise perturbations $[17]$.

Considering that $\mathbf{f}_i$ 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 $\mathbf{f}_{skl} = (f_{skl1}, f_{skl2})$, the error vector $E_i(f_i) = (E_{x_i}, E_{y_i})$ where $x_{skl} = (f_{skl1} - f_{x_i})$ and $y_{skl} = (f_{skl2} - f_{y_i})$ are defined. From this error, the potential function $U_i : \mathcal{R}^2 \rightarrow \mathcal{R}$ is computed as described in Section III-B. Based on these functions, the movement flow for the feature $i$, $\Phi_i$ is defined as a linear combination of two terms

$$\begin{align*}
\Phi_i(f_i) &= \left(\frac{\partial x_i}{\partial y_i} f_i \right) = G_1(f_i) \cdot \left(\frac{\partial G_1_i}{\partial y_i} (\tau) + G_2(f_i) \cdot \left(\frac{\partial U_i}{\partial y_i} \right) \right). \hspace{1cm} (2)
\end{align*}$$

where $G_1, G_2 : \mathcal{M} \rightarrow \mathcal{R}^+$ are weight functions defined in Section III-C, so that $G_1 + G_2 = 1$. The first term in (2), $f_{skl}(\tau)$ is obtained by calculating the Taylor coefficients of the desired trajectory and mimics the behavior of the desired trajectory; therefore, $G_1$ controls the progression speed of the trajectory in the image. The second term in (2) is employed to reduce the tracking error.

As is shown in $[14]$, applying the velocity field codified by $\Phi_i$, the expected evolution for $\beta > 0$ will be

$$\dot{f}_i = \beta \cdot \Phi_i.$$  \hspace{1cm} (3)

So that $E_i$ will vary according to

$$\dot{E}_i = -\beta \cdot G_2(f_i) \cdot \frac{\partial U_i}{\partial E_i}.$$  \hspace{1cm} (4)

Thus, the error evolves in the direction of the negative gradient of the potential and converges at a point of $\mathbf{U}_i$ in which $E_{x_i} = E_{y_i} = 0$, so that $f_i \rightarrow f_{skl}$.

Up to now, a set of $M$ features has been considered, each of which must follow a desired trajectory in the image. However, each of the $M$ trajectories must progress in a coordinated way, so that the shortest trajectories reduce their speed to adapt to the longest ones. The value of $m_i$ for each feature depends on the length of the trajectory in the image, so that the time described for each feature to pass through $N$ points in the image will be the same. Therefore, for a $k$ instant, the set $bkS = \{k, f_{skl} \in 1, \ldots, M\}$ must correspond to a desired configuration of the camera in the 3-D space $k/k, k \in 1, \ldots, N$ (obtained as described in Section II). Therefore, by controlling the progression of each feature $i$ in the image, by means of the parameter $m_i$, the set of features observed $bkS$ is made to belong to the set $S$ defined in Section II.

### B. Potential Function

This section describes the potential function used by the movement flow for one feature. Therefore, for the sake of clarity, the subindex that indicates which feature is being considered, is omitted.

The potential function must attain its minimum when the error $E = f - f_{skl}$ is zero and must increase as $f$ deviates more from its desired location $f_{skl}$ ($f$ being the feature extracted from the image and $f_{skl}$ the feature in $f_{skl}(\tau)$ nearest to $f$). To define the potential function, $I$ is considered as the image that would be obtained after the trajectory $f_{skl}(\tau)$ has been represented [see Fig. 1(a)].

The first step in determining the potential function is to calculate the gradient $I_{\varepsilon}$ of $I$ $[25]$. Once the image $I_{\varepsilon}$ has been obtained, the next step to determine the potential function is to generate a distance map. The distance map creates a distance image $I_d$ of the binary image $I_{\varepsilon}$. That is $[15]$

$$I_d = \Psi_{BC} (I_{\varepsilon}) = \sum_i \varepsilon_{i, BC} (I_{\varepsilon})$$  \hspace{1cm} (5)

where $\varepsilon$ performs the erosion of the image $I_{\varepsilon}$ using the $3 \times 3$ square structuring element $B_C$. The distance map $I_d$ is an image that stores information relative to the shortest distance between each pixel and $f_{skl}(\tau)$. In Fig. 1(b), the distance map for the trajectory in Fig. 1(a), is shown. In this 3D representation, the value of $z$ coordinate represents the distance between each pixel and the nearest pixel to it in the desired trajectory and shows the distribution of the potential function $U$ obtained.

### C. Weight Functions $G_1 - G_2$

From (2), it can be stated that when the tracking error is too high, the value of $G_1$ must be comparatively low with regard to the value of $G_2$, so that the progression speed of the trajectory is reduced and the value of the second term in (2) is increased to be able to quickly reduce the error. On the other hand, if the error is low, the value of $G_2$ must be reduced, and that of $G_1$ must be increased to be able to increase the progression speed of the trajectory. As such, the weights are not dependent on the size or the resolution of the image, but they do depend on the degree of precision required. Therefore, by adjusting the values of $G_1$.
and $G_2$, different types of behavior can be obtained, depending principally on the maximum tracking error permitted.

To determine the values of these weight functions, the following function, which affords a finite value for any value of $\alpha > 0$ is considered:

$$\mu(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}dx. \quad (6)$$

Given two parameters $a > 0$ and $b > 0$, the following exponential distribution is defined for $0 < x < 1$:

$$\nu(x, a, b) = \frac{\mu(a+b)}{\mu(a)\mu(b)}x^{a-1}(1-x)^{b-1}. \quad (7)$$

Based on the functions (6) and (7), $G_1$ and $G_2$ are defined as

$$G_1(f, a, b) = 1 - \nu \frac{U(E(f))}{\delta a b} \mu a^{-1}(1-t)^{b-1}dt \quad (8)$$

$$G_2(f, a, b) = \nu \frac{U(E(f))}{\delta a b} t^{a-1}(1-t)^{b-1}dt \quad (9)$$

the parameter $\delta$ being a variable that represents an error value such that if $U(E(f)) > \delta \rightarrow G_1 = 0$. This variable will also be denoted as the maximum tracking error permitted. The shapes of the functions defined in (8) and (9) are variable, depending on the values of the parameters $a$ and $b$ as illustrated in Fig. 2.

The rationale that underlies the definition of the weights $G_1$ and $G_2$, according to the exponential curve shown in (8) and (9) is the possibility of defining their evolutions by using the parameters $a$ and $b$. To perform a strict tracking of the trajectory, assuring a minimal value of the error, the parameters $a = 0.5$ and $b = 4$ can be used (Fig. 2). As such, the system quickly increases the emphasis on eliminating the image tracking error. On the other hand, if we wish to have more flexible tracking so that the progression speed will be high and only reduces when the error is sufficiently high, the parameters $a = 2$ and $b = 0.5$ can be used (Fig. 2). It should be pointed out that the parameters used for the tests and results presented throughout this article are $a = 2$ and $b = 4$. Thus, considering $a = 2$, $b = 4$, and $\delta = 5$, the vector field that corresponds to the movement flow obtained for the trajectory shown in Fig. 1(a), is represented in Fig. 3. In general, we should state that the values $a$ and $b$ of the weight function do not depend on the proprieties of the desired trajectory, allowing the tracking in any case. Nevertheless, these functions can depend on the task to be developed. Thus, if the task requires greater or lesser precision, the tracking can be more or less precise in modifying the parameters of the weight function, as previously described.

Although, at this point, the determination of the movement flow for the complete image has been described (given that “the region of attraction” covers the entire image), in order to implement the movement flow-based visual servoing strategy, it is only necessary to generate the movement flow for the features that are extracted from the image during the tracking. Therefore, at each iteration, the system only determines the movement flow for four pixels, which correspond to the position in the image of the four features extracted from the visual system.

IV. FUSING FORCE AND VISUAL INFORMATION

In this section, the general strategy for fusing visual and force information and its application to unstructured environments is described. The different phases that compose the fusion strategy are applied to the task shown in Fig. 4. In this task, the robot must maintain flat contact with a constant force being applied between the gripper and the surface. Movement flow-based visual servoing is used so that the desired image trajectory for each feature $f_i(t)$ is the one observed by the camera during the trajectory of the robot (maintaining the contact with the surface). However, this trajectory is subject to errors and, therefore, must be modified according to the information obtained from the force sensor.

Once the contact between the gripper and the object is detected [Fig. 4(a)], the system recognizes the contact surface, correcting possible errors in the orientation of the robot with respect to the object [Fig. 4(b)]. As will be described in this section, in contrast to the use of virtual fixtures, this method allows the robot to know the geometric constraints that exist, without
the need of having any previous knowledge about them. Once
this step has been finished, the system determines the move-
direction nearest to the one obtained from the movement
flow-based visual servoing system and which fulfills the spatial
restrictions imposed by the manipulated object. Once this di-
rection is obtained, the image trajectory, tracked by the move-
ment flow-based visual servoing system, is modified. Thus, the
system is able to track the trajectory, maintaining the contact
with the surface, and obtaining compatible control actions from
both sensory systems [Fig. 4(c)]. Furthermore, we have devel-
oped a system of detecting changes in the model considered
by a Kalman filter applied to the interaction forces obtained.
This system is used not only to detect changes in the surface
[Fig. 4(d)], but also to fuse the visual and force information. We
should also mention that in any application in which the robot
must grasp an object, such operation is done after having per-
formed the complete tracking of the desired trajectory in the
image which, in turn, is coherent with the sensory information
obtained from the force sensor.

In the following sections, each of the steps illustrated in Fig. 4
are described.

A. Recognition of the Contact Surface

The system used for recognizing the surface and orienting
the robot with respect to it only uses the sensor information ob-
tained from the force sensor. As opposed to other methods, such
as [12], [29], in our case, no friction forces are generated during
the recognition. As is shown in [21], the high proportion of noise
in the signal provided by the force sensor and the friction forces
that are generated in practice, can mean that in many environ-
ments, the methods previously employed, such as [12] and [29],
cannot be applied. Following the example shown in Fig. 4(a),
when contact is made, torques are generated in the $x$ and $z$

directions with respect to the robot coordinate frame [see coor-
dinate frame in the upper-right corner of Fig. 4(b)]. These torques
are used to orientate the robot, maintaining a constant force in
the direction that is perpendicular to the contact surface. To do
so, we have used a proportional controller [see (10)] in which
the values of $n_{x,z}$ and $n_{z}$ are zero

\[
\begin{bmatrix}
\omega_x \\
\omega_z
\end{bmatrix} = -\lambda F \cdot \begin{bmatrix}
\frac{n_x - n_{x,z}}{k_{x,x}} \\
\frac{n_z - n_{z,z}}{k_{z,z}}
\end{bmatrix}
\]

(10)

where $n_x$ and $n_z$ are the torques obtained, and $k_{x,x}$ and $k_{z,z}$ the
estimated tool stiffness in each direction. In Fig. 5, the evolution
of the torques obtained until the orientation finishes (and the
torques are, therefore, zero), is shown.

B. Calculating the Desired Trajectory in the Image

Given a collision with the setting and having recognized the
normal vector of the contact surface, as described in the previous
section, the transformation $T$ that the camera must undergo to
fulfill the spatial restrictions, is determined. This transformation
is calculated so that it represents the nearest direction to the one
obtained from the movement flow, and which is contained in the
plane of the surface. Thus, we guarantee that the visual information will be coherent with the sensor information obtained from the force sensor. To do so, considering a feature \( f \) extracted by the camera once the contact surface is recognized, and \([R, t]_e\), a sampling of the transformation \( T \) that the camera undergoes during the tracking of the recognized surface, the feature \( i f \) extracted in each one of these positions will be

\[
i f = A_i f \cdot A_i^{-1} \cdot f + A_i \cdot \frac{t}{z}
\]  

(11)

where \( A \) is the intrinsic parameter matrix and \( z \) is the distance between the camera and the object from which the features are extracted.

From the sampling of the desired trajectory in the image for each feature \( C_i \), we apply a cubic cardinal spline interpolator to obtain the desired trajectory in the image \( f_{i\bar{t}}(\tau) \). From this trajectory, the new movement flow is calculated as described in Section III.

C. Detecting Changes in the Contact Surface

The generalized likelihood ratio (GLR) algorithm [27] applied to the obtained forces is employed to detect changes in the contact surface. This algorithm has been previously employed, for example, in [3] to estimate the movement in an image-based visual servoing task and to detect and compensate this movement. In this section, we apply this method to filter interaction forces and to detect changes in the contact surface.

If the signal obtained from the force sensor is not filtered, the peak-to-peak noise level is about 0.9 N. Due to this high noise level, a low-pass filter with a cutoff frequency at a maximum of 0.19 Hz is required to reduce the maximum difference between the greatest and the lowest forces to 0.18 N. Furthermore, in order to obtain a more reliable measurement, the Kalman filter shown below is applied.

The equations of the state and measurement models of the Kalman filter that allow the estimation of \( x \in \mathbb{R}^n \) from the set of measurements \( k_{x} \) are

\[
k_{x} = F_{x} k_{x} + k_{w}
\]

(12)

and

\[
k_{z} = H_{z} k_{x} + k_{w}
\]

(13)

where \( F \) is the transition state matrix, which relates the state in the previous step \( k = 1 \) to the current state \( k \) without noise, and \( H \) is the measurement matrix, which relates the state to the measurement \( k_{z} \). We assume that the random variables \( k_{w} \) and \( k_{w} \) are independent and with normal distributions (the values of \( Q \) and \( R \) are experimentally determined and are considered to be \( 10^{-5} \) and \( 10^{-2} \), respectively)

\[
p(v) \sim N(0, Q)
\]

(14)

and

\[
p(w) \sim N(0, R).
\]

(15)

When there is a change \( \sigma \) in the surface at a moment \( \theta \) (and, therefore, there is a change in the filtered forces), this change directly influences the innovation value \( k_{\gamma} \) of the Kalman filter, which can be considered as

\[
k_{\gamma} = k_{\gamma} + G(k; \theta) \cdot \sigma
\]

(16)

where \( k_{\gamma} \) represents the innovation that will be obtained if there is no change (i.e., difference between the current measurement \( k_{z} \) and a prediction of it based on previous measurements)

\[
k_{\gamma} = k_{z} - H_{z} \cdot k - 1 \cdot x_{n}
\]

(17)

\( G(k; \theta) \cdot \sigma \) is the effect (on the innovation measured at the iteration \( k \)) of the change in the surface \( \sigma \) that is produced at the iteration \( \theta \).

Thus, the state vector is modified by the change in the surface, in the following way:

\[
k_{x} = k_{x} + J(k; \theta) \cdot \sigma = k_{x} + k_{x}
\]

(18)

where \( k_{x} \) represents the estimation of the state vector obtained if there is no change in the contact surface, \( J(k; \theta) \cdot \sigma \) is the effect (on the value of the estimation of the state vector measured at the iteration \( k \)) of the change in the surface \( \sigma \) that is produced at the iteration \( \theta \), and \( k_{K} \) is the gain of the Kalman filter.

We denote the forces obtained by \( F(f_{x}, f_{y}, f_{z}, n_{x}, n_{y}, n_{z}) \), which compose the state vector. During the task, the forces must remain constant and, therefore, the state model will be

\[
k_{x} = F_{x} k_{x} + k_{w}
\]

(19)

The filter is now applied to detect changes in the surface in a given direction, indicated by the vector \( a = [1 \ 0 \ 0 \ 0 \ 0 \ 0]^{T} \). From (16), the innovation is represented as

\[
k_{\gamma} = g(k; \theta) \cdot \alpha
\]

(20)

where \( g(k; \theta) \) is the value of \( G(k; \theta) \) in the direction of the change \( a \). \( \alpha \) is the size of the hypothetical change in the magnitude of the force.

If \( a = k \), the following values are obtained:

\[
g(\theta; \tau) = G(\theta; \tau) \cdot a = H \cdot a
\]

(21)

\[
j(\theta; \tau) = J(\theta; \tau) \cdot a = k \cdot a
\]

(16)

where \( j(k; \theta) \) is the value of \( J(k; \theta) \) in the direction of the change \( a \). If \( a < k \)

\[
g(\theta; \tau) = G(\theta; \tau) \cdot a = H \cdot f_{x} \cdot J(k; \theta) \cdot a
\]

(22)

\[
j(\theta; \tau) = J(\theta; \tau) \cdot a = F \cdot J(k; \theta) \cdot a + k_{K} \cdot G(k; \theta) \cdot a
\]

(23)
The test function to determine a change in the interaction forces filtered in the direction \( \alpha \) will be

\[
I(k; \theta) = \frac{\partial^2 (k; \theta)}{c(k; \theta)}
\]

(24)

where

\[
c(k; \theta) = \sum_{j=\theta}^{k} g^2(j; \theta) \quad (25)
\]

and

\[
d(k; \theta) = \sum_{j=\theta}^{k} g(j; \theta) \quad \gamma, \quad (26)
\]

This detector determines the presence of a change in the interval \( k - \xi < \theta < k \). The value of \( \xi \) is established high enough to compensate for the noise effect in the signal. We have empirically determined the value \( \xi = 10 \). For any \( \theta \) between \( k \) and \( k - \xi \), the value of \( \theta_m \) that maximizes \( I(k; \theta) \) and that represents the moment at which the change is most probably produced, is determined. Once the test function is defined \( I(k; \theta) \), we define the threshold \( U \) so that if \( I(k; \theta_m) > U \), then a change in the surface is produced (empirically, this threshold has been established at 2000).

In Fig. 6, the values of the force \( f_x \) and the parameter \( I(k; \theta_m) \) obtained during the tracking of two different surfaces are shown [the experimental setup shown in Fig. 10(a) is employed]. It can be observed that by using the threshold previously defined, the system can automatically detect the change in the surface.

### D. Strategy for Fusing Visual and Force Information

Up to now, the majority of approaches for fusing visual and force information are based on hybrid control. Only recently [2] has it been possible to find studies on the control of a given direction using force and vision simultaneously (shared control). These approaches are based on the “task frame” formalism [4], [8]. These works suppose the presence of a high-level descriptor of the actions to be carried out in each direction of the workspace at each moment of the task. Thus, the geometric properties of the environment must be known previously. The approach described in this section does not require specifying the sensory systems to be used for each direction. Furthermore, the proportion of information used from each sensor depends on the criterion described in this section.

From the system defined in the previous section, the parameter \( I(k; \theta) \), which provides information about whether the tracking is correctly developed, is obtained. We have developed an empirical study and we have defined the following thresholds:

- \( U_1 \). Normal functioning of the system. This is the range of values of \( I(k; \theta) \) that can be considered as normal during the tracking.
- \( U_2 \). Change in the surface. This threshold is the same as the one defined as \( U \) in Section IV-C, so that if the value of \( I(k; \theta) \) is greater than this threshold, it is assumed that there is a change in the contact surface and the robot must therefore be oriented in relation to the new surface (see Section IV-A).
- \( U_3 \). Upper limit of incorrect functioning. This threshold is between the thresholds \( U_1 \) and \( U_2 \), and is defined to characterize the highest values of \( I(k; \theta) \) that can be obtained when there is not a change in the surface.

When the value of \( I(k; \theta) \) increases, this can be due to irregularities or changes in the contact surface. However, we have determined that this increase can be also obtained when, for several possible reasons (irregularities in the surface, errors in the trajectory generated by the visual servoing system, high velocity established by the movement flow, etc.), the tracking is not correctly done and, therefore, the system cannot maintain a constant force on the surface. The behavior is then more oscillatory, and changes are generated in the interaction forces, increasing the value of \( I(k; \theta) \). To correct this behavior, the proportion of information used from the force sensor can be augmented when the value of \( I(k; \theta) \) increases, as described below.

The final control action \( \theta^{C} \) will be a weighted sum obtained from the movement flow-based visual servoing system \( \theta^{C} \), and from the force sensor \( \theta^{F} \).

\[
\theta^{C} = \frac{1}{\lambda_{F}} \cdot (\mathbf{F} - F(d)) / k \quad \text{so that} \quad \theta^{C} = \lambda_{F} \cdot \eta^{C}_{\mathbf{M}} + \eta^{C}_{\mathbf{F}}. \quad (27)
\]

Thus, depending on the value of \( I(k; \theta) \) and the previously defined thresholds, we obtain the following control actions:

- \( I(k; \theta) < U_1 \). Normal functioning of the system. In this case, both control actions are weighted with the same proportion

\[
\eta^{C} = 0.5 \cdot \eta^{C}_{\mathbf{M}} + 0.5 \cdot \eta^{C}_{\mathbf{F}}. \quad (27)
\]

- \( U_1 \leq I(k; \theta) < U_3 \). Range of values of \( I(k; \theta) \) that can be obtained when a change in the surface begins or when the system works incorrectly. In this case, the weight applied to the control action corresponding to the movement flow-based visual servoing system is reduced with the aim of correcting defects in the tracking. Before describing the weight function for this range of \( I(k; \theta) \), two parameters that characterize this function are defined. These parameters \( (p_1, p_2) \) identify the velocity range that the movement flow-based visual servoing system can establish for different values of \( I(k; \theta) \). Thus, when \( I(k; \theta) \) is equal to \( U_1 \), or lower, the velocity established by the computer...
vision system will be 
\[ \mathbf{v}_{\text{Min}}^c = -\left(\lambda_M/2\right) \cdot \mathbf{J}_f^T \cdot \mathbf{e}_r, \]
that is to say, the normal velocity defined to carry out the tracking of the trajectory in the image space. In the previous expression, we can see the term \(\lambda_M/2\) due to the weight in the control action obtained from the computer vision system \(\mathbf{v}_M^c\), in the global control action \(\mathbf{v}_G^c\), that is to say, \(p_1 = 0.5\) \([27]\). However, when \(l(k;\theta)\) is equal to \(U_2\), we define \(\mathbf{v}_{\text{Min}}^c = -\lambda_M \cdot p_2 \cdot \mathbf{J}_f^T \cdot \mathbf{e}_r < \mathbf{v}_{\text{Min}}^c\) as the minimum velocity, empirically obtained, to carry out the tracking of the trajectory and which allows the system to correct the possible defects in this trajectory (the effect of the force control in the trajectory is increased in the global control action). Thus, the value of the weight associated with the velocity provided by the visual servoing system, will be \(p_2 = 0.5 \cdot (\mathbf{v}_{\text{Min}}^c/\mathbf{v}_{\text{Max}}^c)\). Therefore, considering a decreasing evolution of the weight function applied to the velocity obtained from the movement flow-based visual servoing system, this function will have the following value in the range \(U_1 \leq l(k;\theta) < U_3\):

\[
p_F = \frac{p_2 - p_1}{U_3 - U_1} \cdot l(k;\theta) + p_1 - \frac{p_2 - p_1}{U_3 - U_1} \cdot U_1, \tag{28}\]

Obviously, the weight associated with the force control system will be \(p_F = 1 - p_r\).

- \(U_3 \leq l(k;\theta) < U_2\). Security margin. When \(l(k;\theta)\) is in this range, the behavior established is to continue with the minimum velocity \(\mathbf{v}_{\text{Min}}^c\).

- \(l(k;\theta) \geq U_2\). In this case, a change in the surface is made and, therefore, to orientate the robot with respect the new surface, the concepts described in Section IV-A are applied.

V. RESULTS

For the tests we have used, an eye-in-hand camera system composed of a JAI-M536 mini-camera in the end-effector of a 7 d.o.f. Mitsubishi PA-10 robot [Fig. 7(a)] also equipped with a force sensor (67M25A-I40 from JR3, Inc.). MATROX GENESIS is used as the image acquisition and processing board. In the experiments described in this paper, the tracked target is composed of four gray marks [Fig. 10(a)]. The visual features extracted are the image coordinates of the center of gravity of each mark, which are adequate for controlling the task [5].

A. Tracking Trajectories

This first experiment consists of the tracking of a parabolic trajectory. Fig. 7(b) and (c) show a sampling of the desired Cartesian trajectory followed by the camera in the 3-D space, together with the corresponding sampling of the desired trajectory in the image \(k_{\text{f_i}}\). Fig. 7(b) shows the evolution of the coordinate frame at the end-effector of the robot during the desired trajectory.

To obtain the movement flow, the parameters of (8) and (9) have been considered as \(b = 10\), \(a = 2\), and \(b = 4\). Once the above-mentioned movement flow has been determined, the trajectories of the features in the image, shown in Fig. 8(a), are obtained using movement flow-based visual servoing.

From the trajectory obtained in Fig. 8(a), it can be concluded that the existing error between the desired and the obtained trajectories is confined within the limits expected from the parameters used for obtaining the movement flow. Fig. 8(b) shows the desired trajectory and the one obtained in the 3-D Cartesian space.

B. Movement Flow-Based Visual Servoing System versus Time-Dependent Tracking Systems

The systems that use image-based visual servoing to track trajectories that have been employed up to now [17] use a time-variable reference. These systems do not guarantee the correct tracking of the trajectory because the references can be very restrictive and, therefore, the system tries to maintain the time-references even if the tracking is not carried out correctly. This problem is solved by using the movement flow-based visual servoing, because this system is not affected by time restrictions.

A new experiment, in which the robot must interact with objects within the workspace and, therefore, can be obstructed during a certain time, is now shown to illustrate the difference between the two systems. In such a situation, a time-dependent tracking system based on visual servoing has the behavior shown in Fig. 9(a). This figure shows the 3-D trajectory of the robot end-effector during the task. Due to the obstruction, the references are delayed and, therefore, are very restrictive, not allowing the correct tracking. However, using movement flow-based visual servoing [Fig. 9(b)], once the obstruction has ended, the system continues with the tracking and is not affected by the delay.
C. Fusing Force and Visual Information

This section shows two experiments that validate several aspects of the proposed algorithms. The first test consists of carrying out the tracking of a curved surface. The experimental setup shown in Fig. 10(a) is employed. Applying the algorithm for fusing visual and force information described in Section IV, the 3-D trajectory represented in Fig. 10(b) is obtained. In this figure, it can be observed that the robot remains perpendicular with respect to the surface during the trajectory.

To verify that the system can detect the change in the surface in Fig. 11, the interaction forces obtained (in z direction) and the value of $I(\kappa; \beta)$ are represented. As previously indicated, the threshold used for detecting changes in the surface is 2000, so that the system is able to detect the change.

Fig. 13(a) shows a new experiment in which the gripper has an incorrect orientation to grasp the object. The only limitation to the grasping of the object is its size. As we have been describing throughout this article, the visual-force control fusion algorithm proposed here allows the robot to adapt to the irregularities of the surfaces of the object being manipulated so that such surfaces need not be flat. This algorithm allows the modification of the tracking speed, depending on the irregularities or any other changes in the surfaces. Furthermore, in this experiment, the gripper collides with the object during the trajectory, so that, following the algorithm previously presented, the robot must orientate the gripper with respect to the object, to determine the new trajectory in the image space [see the marks in Fig. 13(a) that will be the features extracted from the image]. Only after the trajectory desired in the image has been completely tracked is the object grasped, permitting the robot to obtain the desired contact points when the correct tracking of the trajectory in the image is performed [Note that this is generated from the 3-D trajectory to be tracked $\gamma(t)$ (see Section II)].

In this case, the axis with respect to which the orientation is carried out does not intersect the z axis of the robot (which is the same axis as the force sensor). Therefore, it is necessary to transform the obtained forces with respect to the collision point frame. We have determined that the correct orientation of the gripper is obtained when the torque with respect to the
rotation axis of the robot $n_x$ is lower than $-2$ N m. In Fig. 12(a), the values of $n_x$ for several experiments are shown. Fig. 12(b) shows that the system allows a constant force to be maintained on the contact surface until the object can be grasped.

In Fig. 13(b) and (c), a 3-D representation of the positions obtained during the task is shown.

VI. CONCLUSION

A new method for fusing sensor information from a computer vision system with that from a force sensor has been described. The method was applied to different manipulation tasks that required great precision. In this application, it is not only necessary to obtain given features in the image from the initial features, but a tracking of the desired trajectory between them, fulfilling the desired spatial restrictions is also necessary for the correct development of the task.

A method for the tracking of trajectories, called movement flow-based visual servoing, has been described. This method avoids the use of information about time to ensure correct tracking, thus overcoming the limitations of the tracking systems based on visual servoing that have, so far, been proposed.
The system’s different parameters, which allow the regulating of the desired precision and speed during the tracking, have also been presented.

The sensor fusion method employs information from the movement flow-based visual servoing system, which affords, as the results show, the correct behavior of the tracking system, not only in the image space but also in the 3-D space.

REFERENCES


Jorge Pomares received the Computer Science Engineer and Ph.D. degrees from the University of Alicante, Alicante, Spain, in 1999 and 2004, respectively. Currently, he is a Professor in the Department of Physics, Systems Engineering, and Signal Theory of the University of Alicante, where he teaches courses in robotics and automatic control. He also conducts research in the Automatics, Robotics and Computer Vision Group in the same department, where he has been involved in several research projects in collaboration with industry and the Spanish Government. His main research interests are visual servoing, robotics, and computer vision. Dr. Pomares is a member of the CEA-IFAC.

Fernando Torres (M’03) received the industrial engineer and Ph.D. degrees from the Polytechnic University of Madrid (UPM) in 1991 and 1995, respectively. Currently, he is a Professor in control, robotics, and computer vision at Alicante University, where he has been since 1994. His research interests include automatic visual inspection, robotics, manufacturing automation, visual servoing, morphological processing, and new technologies for teaching. He has published several international papers. Dr. Torres is a member of the IFAC TC 5.1, CEA-IFAC, and the Spanish Image Analysis and Pattern Recognition Society.