International Conference on Robotics and Automation
Barcelona, Spain, April 2005

Time Independent Tracking Using 2-D Movement Flow-Based Visual Servoing*

Jorge Pomares, Fernando Torres

Department of Physics, System Engineering and Signal Theory
University of Alicante
P. O. Box 99 03080 Alicante, Spain
{jpomares, Fernando.Torres}@ua.es

***Abstract* - In this paper, the so-called 2-D movement flow-based visual servoing system is proposed, which allows the tracking of trajectories planned in the image space. This method is time-independent, which is useful when problems such as obstructions occur during the tracking. This method allows the tracking of image trajectories assuring the correct behaviour in the 3-D space and avoiding the limitations of the time-dependent systems used up to now for tracking trajectories. This aspect has allowed its application to manipulation tasks in which the trajectory can be obstructed during the tracking.**

Index Terms* - *2-D visual servoing, positioning tasks, vector field, eye-in-hand system, tracking trajectories.

I. Introduction

The visual servoing systems have been used so far in a wide range of applications, the majority of which are point to point-based, in which the objective is to achieve a desired configuration of the features in the image from an initial one. As several previous works have shown [1], if the initial and final configurations are very far apart, their convergence is difficult to ensure, particularly if there are modelling errors.

Typically, the visual servoing systems are position-based and image-based classified [2]. The position-based visual servoing [3] requires the computation of a 3-D Cartesian error for which a perfect CAD-model of the object and a calibrated camera is necessary. These types of systems are very sensitive to modelling errors and noise perturbations. In image-based visual servoing [1] the error is measured directly in the image. This approach ensures the robustness with respect to modelling errors but generally an inadequate movement of the camera in the 3-D Cartesian space is obtained. On the other hand, it is well known that image-based visual servoing is locally stable, which ensures a correct convergence if the desired configuration is sufficiently near the current one. As is shown in [4] a possible solution is the use of a planning step jointly with the servoing one so that the tracking error along the planned path always remains small.

This paper presents a solution for tracking image trajectories and, to confirm the validity of the approach, an eye-in-hand robotic system is applied. Since we are not interested in image processing issues in this paper the tracked target is composed by four grey marks whose centres will be the extracted features during the tracking. The approach described in this article uses what we call a movement flow to determine the desired configuration from the current one. As such, the desired configuration remains close to the current one and the local stability of the image-based visual servoing is assured, allowing a tracking of the desired trajectory in the image. Only recently has it been possible to find studies related to the use of visual servoing systems to follow trajectories in the image [5]. To do so, an approach that uses an image-based visual servoing system to track a desired timed trajectory is presented. In contrast to the approach described in [5], the one described in this article is employed to track non-timed trajectories, using movement flow. If the tracking is formulated as a timed trajectory in the image, the current configuration and the desired one are separated by a time interval, Δt. 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 IV-C). To resolve this problem, the so-called movement flow-based visual servoing is used.

The movement flow is a vector field that is used by an image-based controller to determine the location of the desired features in the image and consequently allowing the tracking of the desired trajectory. The task to be carried out by the robot is encoded in the image configuration space, i.e., specifying a desired configuration $\mathbf{s}_d=\{\boldsymbol{f}_1+m_1\Phi_1(\boldsymbol{f}_1), \boldsymbol{f}_2+m_2\Phi_2(\boldsymbol{f}_2),\ldots, \boldsymbol{f}_M+m_M\Phi_M(\boldsymbol{f}_M)\}$ at each current possible configuration $\mathbf{s}=\{\boldsymbol{f}_1, \boldsymbol{f}_2,\ldots, \boldsymbol{f}_M\}$, where $\boldsymbol{f}_i$ / $i\in 1\ldots M$ are the features extracted in the image, Φ_i is the movement flow for the feature i, and $\boldsymbol{m}=\{m_1, m_2,\ldots, m_M\}$ determines the progression speed.

This paper is organized as follows: Section II describes the basic concepts used about path planning in the image space. Section III shows how the movement flow is generated from the trajectory in the image which was obtained in Section II. In Section IV, real experimental results, using an eye-in-hand camera system, confirm the validity of the control scheme proposed. The final section presents the main conclusions arrived at.

II. Path Planning in the Image

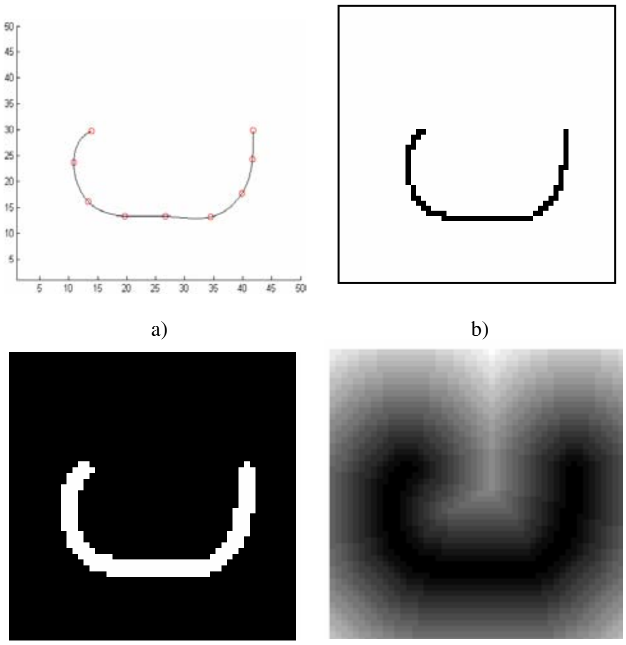
To carry out the path planning of the trajectory in the image, first the discrete 3-D camera path is performed as a sequence of N intermediate camera poses. To realize this first phase, the algorithms developed in our previous studies [6],[7] are employed. Basically, these algorithms consist of the determination of the robot's movements using the solutions of a set of linear programming problems obtained

[*] This work is partially supported by the Spanish MCYT project "DESAURO: Desensamblado Automático Selectivo para Reciclado mediante Robots Cooperativos y Sistema Multisensorial" (DPI2002-02103).

from an object-oriented model. This trajectory guarantees that the target is observed by the camera during the entire trajectory and, therefore, the features remain in the field of view. Employing these algorithms, the discrete trajectory of the target in the image is obtained. This discrete trajectory in the image is composed of M trajectories, $\boldsymbol{F}_{\mathrm{i}} \,/\, \mathrm{i} \in 1...\mathrm{M}$ (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 $\boldsymbol{F}_{\mathrm{i}} = \left\{ {}^{\mathrm{k}}\boldsymbol{f}_{\mathrm{i}} \,/\, \mathrm{k} \in 1...\mathrm{N} \right\}$. This will be the sampling of the desired trajectory in the image that must be tracked by the movement flow-based visual servoing system.

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 a tracking of the trajectory at each instant. When the tracking of a trajectory is required, the fact that a given point is reached at any given moment does not offer any information about the precision of the tracking. According to this principle, for each feature, a codification of the desired trajectory in the image $\boldsymbol{f}_{\mathrm{id}}(\tau)$ is done, so that, $\boldsymbol{f}_{\mathrm{id}}$:$\Gamma \rightarrow \Im$ where $\Gamma \subset \Re$, such that if Γ was the time, then the classical timed-trajectory in the image would be obtained. For this reason, the desired trajectory, $\boldsymbol{f}_{\mathrm{id}}(\tau)$, is codified, for each feature, using a cubic cardinal spline interpolator (applied to the samples of $\boldsymbol{F}_{\mathrm{i}}$) and a representation of this trajectory in the image space is obtained (see Fig. 1.a and Fig. 1.b).

Next, for each trajectory, the Taylor coefficients up to the second order at each pixel of the trajectory defined in $\boldsymbol{f}_{\mathrm{id}}(\tau)$ is determined. At every point of the trajectory, the third and fourth coefficient form a vector that is tangent to the trajectory to be tracked $\left(\dot{f}_{\mathrm{xid}}, \dot{f}_{\mathrm{yid}}\right)$. These values will be used to determine the movement flow.



a)

b)

c)

d)

Fig. 1 a) Desired trajectory in the image space, $\boldsymbol{f}_{\mathrm{d}}(\tau)$, for a feature. b) Trajectory in the image. c) Gradient image. d) Distance map.

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-B. First however, in Section III-A, the potential function used is detailed. Finally, the weight functions employed are described in Section III-C.

A. Potential Function

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

The potential function **U** can be used to measure the tracking performance. Therefore, the potential function must attain its minimum when the error, $\boldsymbol{E}$=$\boldsymbol{f}$-$\boldsymbol{f}_{\mathrm{d}}$, is zero and must increase as $\boldsymbol{f}$ deviates more from its desired location $\boldsymbol{f}_{\mathrm{d}}$, ($\boldsymbol{f}$ being the feature extracted in the image and $\boldsymbol{f}_{\mathrm{d}}$ the feature in $\boldsymbol{f}_{\mathrm{d}}(\tau)$ nearest to $\boldsymbol{f}$). To define the potential function, **I** is considered as the desired image, i.e., the image that would be obtained after the trajectory $\boldsymbol{f}_{\mathrm{d}}(\tau)$ has been represented.

The first step in determining the potential function is to calculate the gradient of **I**. To do so, the combination of the mathematical morphological operations that consist of the subtraction of the erosion, ε, from the dilation, δ, of the image **I** is used [8]. In both the erosion and the dilation a 3x3 square structuring element is used. As result of these operations the image $\mathbf{I}_{\mathrm{g}}$ is obtained:

$$\mathbf{I}_{\mathrm{g}} = \psi\left(\mathbf{I}\right) = \delta\left(\mathbf{I}\right) - \varepsilon\left(\mathbf{I}\right) \qquad (1)$$

In Fig. 1.b the image **I** is shown, which represents the desired trajectory of a feature. This trajectory corresponds to the one represented in Fig. 1.a. In Fig 1.c the image resulting from the calculation of the gradient, $\mathbf{I}_{\mathrm{g}}$, is shown.

Once the image $\mathbf{I}_{\mathrm{g}}$ has been obtained, the next step to determine the potential function is to generate a distance map. The distance map creates a distance image $\mathbf{I}_{\mathrm{d}}$ of the binary image $\mathbf{I}_{\mathrm{g}}$. The value of $\mathbf{I}_{\mathrm{d}}$ at the pixel x is the Euclidean distance of x from the complement of $\mathbf{I}_{\mathrm{g}}$ [9]. That is, the distance of x from the nearest point in the desired trajectory:

$$\mathbf{I}_{\mathrm{d}} = \Psi_{B_{\mathrm{C}}}\left(\mathbf{I}_{\mathrm{g}}\right) = \sum_{i} \varepsilon_{i,B_{\mathrm{C}}}\left(\mathbf{I}_{\mathrm{g}}\right) \qquad (2)$$

where ε performs the erosion of the image $\mathbf{I}_{\mathrm{g}}$ using the 3x3 square structuring element B_{C}. The distance map $\mathbf{I}_{\mathrm{d}}$ is an image that stores information relative to the shortest distance between each pixel and $\boldsymbol{f}_{\mathrm{d}}(\tau)$. In Fig. 1.d, the distance map for the example trajectory in Fig. 1.a, is shown. Based on this figure, it is possible to generate a three-dimensional representation of the distance map (Fig. 2), in which the value of *z* coordinate represents the distance between each pixel and the nearest pixel to it in the desired trajectory. In this figure, it can be seen that, for each pixel in the image, the nearest pixel to it in the desired trajectory can be determined by a gradient descent, avoiding local minima. Based on the representation of the distance map in Fig. 2, equi-potential lines can be determined, which shows the distribution of the potential function **U** obtained.

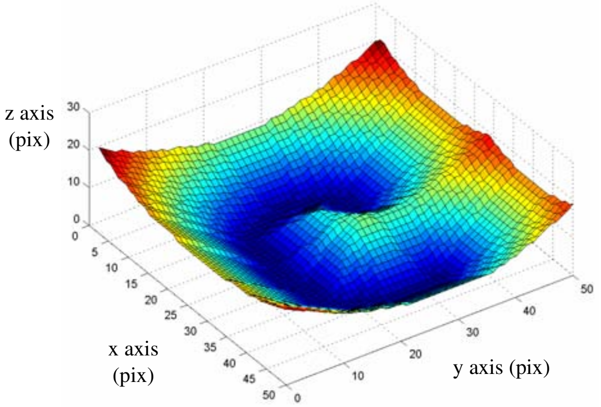


Fig. 2. Three-dimensional representation of the distance map

From **U** the x and y components of the two-dimensional numerical gradient, which can be considered a matrix of vectors pointing in the direction of increasing values of **U,** is obtained. Therefore, if $-\nabla\mathbf{U}$ is represented, a vector field which directs each pixel in the image towards the pixel in $\boldsymbol{f}_{\mathrm{d}}(\tau)$ that is nearest to it, is obtained.

B. Definition

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:

$$\boldsymbol{v}^{\mathrm{C}} = -\lambda \cdot \hat{\mathbf{J}}_{\mathrm{f}}^{+} \cdot \boldsymbol{e}_{f} \tag{3}$$

where λ is the gain of the proportional controller; $\hat{\mathbf{J}}_{\mathrm{f}}^{+}$ is an estimation of the pseudo-inverse of the interaction matrix [2], $\boldsymbol{e}_{f} = \mathbf{s} - \mathbf{s}_{\mathrm{d}}$, $\mathbf{s}=[\boldsymbol{f}_1, \boldsymbol{f}_2,\ldots, \boldsymbol{f}_{\mathrm{M}}]^{\mathrm{T}}$ are the features extracted from the image and $\mathbf{s}_{\mathrm{d}}=[\boldsymbol{f}_1+m_1\Phi_1(\boldsymbol{f}_1), \boldsymbol{f}_2+m_2\Phi_2(\boldsymbol{f}_2),\ldots, \boldsymbol{f}_{\mathrm{M}}+m_{\mathrm{M}}\Phi_{\mathrm{M}}(\boldsymbol{f}_{\mathrm{M}})]^{\mathrm{T}}$, therefore, by modifying the value of $\boldsymbol{m}$ a faster or slower tracking is obtained (see Section IV-A).

Considering that $\boldsymbol{f}_{\mathrm{i}}$ 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 $\boldsymbol{f}_{\mathrm{id}}=(f_{\mathrm{xid}}, f_{\mathrm{yid}})$, the error vector $\boldsymbol{E}_{\mathrm{i}}(\boldsymbol{f}_{\mathrm{i}})=(E_{\mathrm{xi}}, E_{\mathrm{yi}})$ where $E_{\mathrm{xi}}=(f_{\mathrm{xi}}\text{-}f_{\mathrm{xid}})$ and $E_{\mathrm{yi}}=(f_{\mathrm{yi}}\text{-}f_{\mathrm{yid}})$ is defined. From this error the potential function $\mathbf{U}_{\mathrm{i}}:\Re^{\mathrm{n}} \rightarrow \Re$ is computed as described in Section III-A. The more $\boldsymbol{f}_{\mathrm{i}}$ deviates from the desired location $\boldsymbol{f}_{\mathrm{id}}$, the more $\mathbf{U}_{\mathrm{i}}$ increases and, therefore, attains its minimum when the error is zero.

Based on the above-mentioned functions, the movement flow for the feature i, Φ_{i}, is defined as a linear combination of two terms:

$$\Phi_{\mathrm{i}}\left(\boldsymbol{f}_{\mathrm{i}}\right) = G_1\left(\boldsymbol{f}_{\mathrm{i}}\right) \cdot \begin{pmatrix} \dfrac{\partial f_{\mathrm{xid}}(\tau)}{\partial\tau} \\ \dfrac{\partial f_{\mathrm{yid}}(\tau)}{\partial\tau} \end{pmatrix} - G_2\left(\boldsymbol{f}_{\mathrm{i}}\right) \cdot \begin{pmatrix} \dfrac{\partial \mathbf{U}_{\mathrm{i}}}{\partial E_{\mathrm{xi}}} \\ \dfrac{\partial \mathbf{U}_{\mathrm{i}}}{\partial E_{\mathrm{yi}}} \end{pmatrix} \tag{4}$$

where G_1, G_2: $\Im \rightarrow \Re^{+}$ are weight functions defined in Section III-C, so that $G_1 + G_2 = 1$. The first term in (4), $\dot{\boldsymbol{f}}_{\mathrm{id}}(\tau)$, is obtained by calculating the Taylor coefficients of the desired trajectory, as described in Section II.

As is shown in [10], applying the velocity field codified by Φ_{i}, the expected evolution for $\beta > 0$ will be:

$$\dot{\boldsymbol{f}}_{\mathrm{i}} = \beta \cdot \Phi_{\mathrm{i}} \tag{5}$$

So that $\boldsymbol{E}_{\mathrm{i}}$ will vary according to:

$$\dot{\boldsymbol{E}}_{\mathrm{i}} = -\beta \cdot G_2\left(\boldsymbol{f}_{\mathrm{i}}\right) \cdot \frac{\partial \mathbf{U}_{\mathrm{i}}}{\partial \boldsymbol{E}_{\mathrm{i}}} \tag{6}$$

Thus, the error evolves in the direction of the negative gradient of the potential and converges at a point of $\mathbf{U}_{\mathrm{i}}$ in which $E_{\mathrm{xi}}=E_{\mathrm{yi}}=0$, so that $\boldsymbol{f}_{\mathrm{i}} \rightarrow \boldsymbol{f}_{\mathrm{id}}$.

A separate consideration must be realized for the speed of progression. 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. Therefore, for a k instant, the set ${}^{\mathrm{k}}\mathbf{s} = \left\{ {}^{\mathrm{k}}\boldsymbol{f}_{\mathrm{i}} / \mathrm{i} \in 1 \ldots \mathrm{M} \right\}$ must correspond to a desired configuration of the camera in the 3-D Cartesian space (obtained as described in Section II).

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. Furthermore, if M desired trajectories of the same lengths are considered, at any given instant each of the M features must compose a desired configuration. If the following features $\left\{ {}^{\mathrm{k}}\boldsymbol{f}_1 \quad {}^{\mathrm{k}}\boldsymbol{f}_2 \quad \cdots \quad {}^{\mathrm{k+j}}\boldsymbol{f}_{\mathrm{i}} \quad \cdots \quad {}^{\mathrm{k}}\boldsymbol{f}_{\mathrm{M}} \right\}$ are obtained, the value of m_{i} will be: $m_{\mathrm{i}} = m_{\mathrm{i}} \cdot exp\left(\varepsilon_{\mathrm{m}}, \mathrm{j}\right)$ with ε_{m} being the maximum difference allowed between k+j and k, and exp an exponential decreasing function so that exp=1 if j = 0 and exp=0 if j≥ε_{m}. As Section IV shows, the results obtained indicate a good behaviour of the system in the 3-D space when low values of ε_{m} are considered.

C. Weight Functions G_1 – G_2

As can be seen in (4), the first component of the movement flow mimics the behaviour of the desired trajectory, and, therefore, G_1 controls the progression speed of the trajectory in the image. The purpose of the second term is to reduce the tracking error, and therefore G_2 controls the strength of the gradient field. As such, adjusting the values of G_1 and G_2 we can determine how the progression of the trajectory in the image will be, and also control the reduction of the tracking error.

As previously shown, the potential function $\mathbf{U}(\boldsymbol{E})$ can be used to measure the tracking performance. As such, when the tracking error is too high the value of G_1 will 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 (4) is increased to be able to quickly reduce the error. On the other hand, if the error is low, the value of G_2 will be reduced, and that of G_1 will be increased to be able to increase the progression speed of the trajectory. Specifically, these weight functions are defined as a family of curves that are non-zero only in the interval (0,1) (see Fig. 3). The rationale that underlies the definition of the weights G_1 and G_2, according to these exponential curves is the possibility of defining their evolutions by using the parameters a, b and δ, being δ a variable that represents an error value such that if $\mathbf{U}\left(\boldsymbol{E}\left(\boldsymbol{f},\tau\right)\right) > \delta \rightarrow G_1 = 0$. This variable will also be denoted as the maximum tracking error permitted.

To realize a strict tracking of the trajectory, assuring a minimal value of the error, the parameters a=0,5 and b=4 can be used (see Fig. 3). As such, the system quickly increases the emphasis on eliminating the image tracking error. On the other hand, if we wish to have a 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. It should be pointed out that the parameters used for the tests and the results presented throughout this article are a=2 and b=4. Fig. 4 shows the movement flow obtained for the trajectory represented in Fig. 1.a, using the potential function described in Section III-A, together with the above-mentioned values of G_1 and G_2, as well as δ=5.

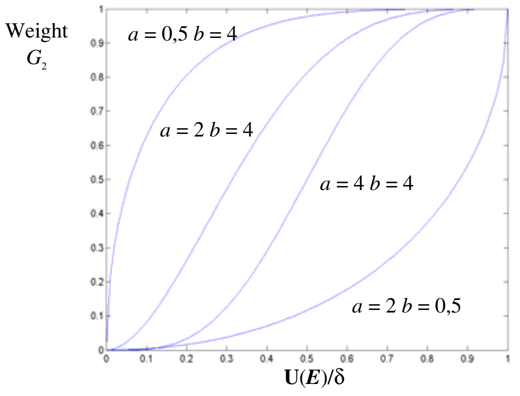


Fig. 3. Evolution of G_2 using different values of a and b.

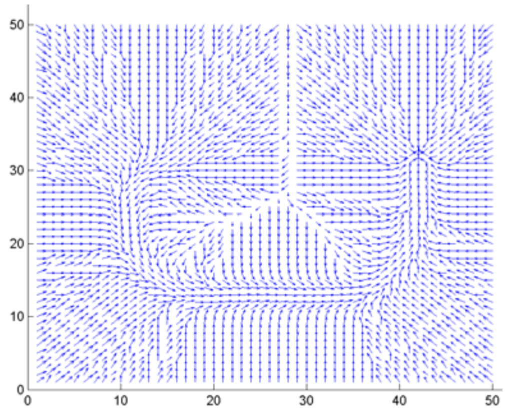


Fig. 4. Movement flow for the trajectory shown in Fig. 1.a.

IV. Results

In this section, we describe the different tests carried out that show the correct behaviour of the system. The capturing of images from the robot end-effector (a 7 d.o.f. Mitsubishi PA-10 robot) is carried out with a JAI-M536 camera. 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). The tracked target has four marks which will be the features extracted during the tracking.

A. *Trajectory 1*

This first experiment consists of the tracking of a parabolic trajectory between the points (327, -330, -82) and (0, -130, 0), expressed in millimetres. Fig. 5 shows a sampling of the desired Cartesian trajectory which must be followed by the camera in the 3-D space, together with the corresponding sampling of the desired trajectory in the image, $\boldsymbol{F}_i = \left\{ {}^{k}\boldsymbol{f}_i \,/\, k \in 1...N, i \in 1...M \right\}$.

To obtain the movement flow, the parameters of the weight Functions $G_1 - G_2$ are δ=10, a=2 and b=4. Once the above-mentioned movement flow have been determined, it is necessary to adjust the speed of progression, $\boldsymbol{m}$. To do so, a study of the behaviour of the system when the value of m_i is modified is now shown (In this study, the same value of m_i=m is considered for each feature). To analyze the behaviour of the system in the face of variations in m, in Fig. 6 the trajectory obtained in the image when the value of m increases, is shown. In this figure, it can be observed that when the progression speed increases, the obtained error increases (see the zones marked in Fig. 6 in which the error is more evident). Likewise, the time required to describe the trajectory diminishes (see Table I). It can be concluded that the value of m that allows a correct tracking of the trajectory in an acceptable time is 20.

It is necessary to demonstrate the correct behaviour of the system, not only in the image, but also in the 3-D space. As such, a comparison between the desired trajectories and the ones obtained in the image and in the 3-D space is shown in Fig. 7. From this figure it can be concluded that the existing error between the desired and the obtained trajectories is confined within the limits expected, based on the parameters used for calculating the movement flow, thus obtaining a correct behaviour in both spaces.

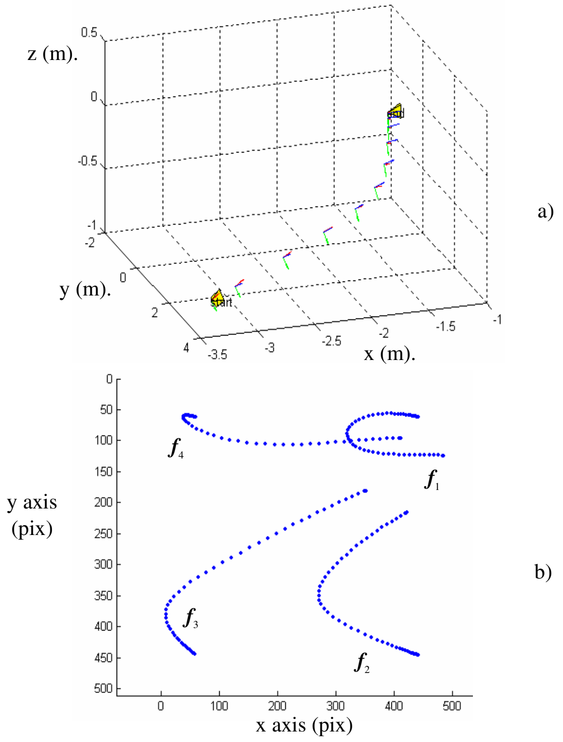


Fig. 5. Sampling of the desired trajectory in the Cartesian space and in the image (Trajectory 1).

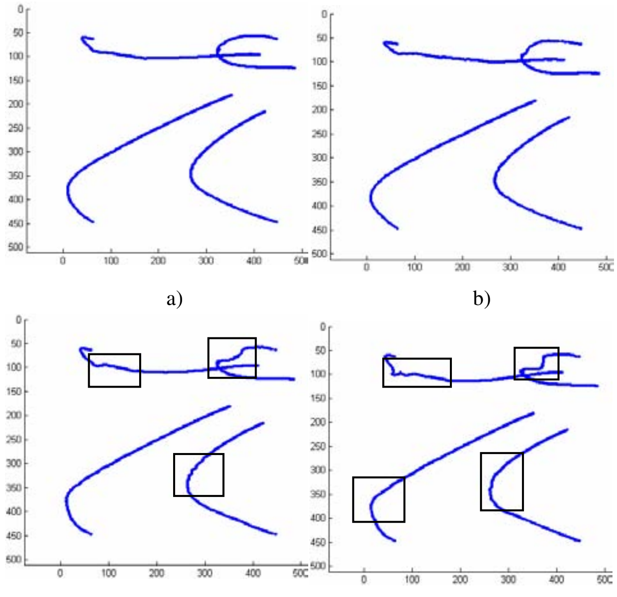


Fig. 6. Results obtained for the tracking of trajectories by modifying the progression speed. a) *m*=10. b) *m*=20. c) *m*=30. d) *m*=40.

To carry out a good tracking of the desired trajectory in the case that "loops" appear in the image trajectory, a trajectory control system has been developed. At each moment, this system determines the segment of the desired trajectory that is closest to the feature extracted from the image and which must be followed. Once this segment has been determined, only the movement flow for this segment of the desired trajectory is calculated. Developing this trajectory control system two advantages are obtained. It not only allows making the tracking when loops appear in the trajectory, but also it accelerates the times required to determine the movement flow since it is only necessary to determine the movement flow closest to the location of the feature in the image (Using a 2,86GHz Pentium 4 the system takes 1.349 msec. in computing the movement flow). Nevertheless, the necessity to determine the surroundings closest to the feature in the image in which the movement flow will be calculated implies the necessity to make an estimation of the movement described by the feature in the image. To do so an estimator based on the Kalman filter has been developed whose details are described in our previous studies [11].

B. *Trajectory 2*

The behaviour of the system proposed for a more complex trajectory of the camera in the 3-D space will now be evaluated (the initial and the final position in millimetres of the robot end-effector are (0, 710, 750) and (-150, 680, 550) respectively). This trajectory is represented in Fig. 8.a. In this case, for the sake of clarity, only the position of the robot end-effector is represented and not its associated coordinate frame. During this 3-D trajectory, the evolution of the features in the image represented in Fig. 8.b is obtained. To carry out this test, the same parameters that we have used in the previous cases are employed. Once again, the correct behaviour of the system, not only in the image but also in the 3-D space, is observed.

TABLE I
TIMES REQUIRED FOR TRACKING THE TRAJECTORY

m	Time (sec.)
10	11.09
20	5.97
30	4.17
40	3.32

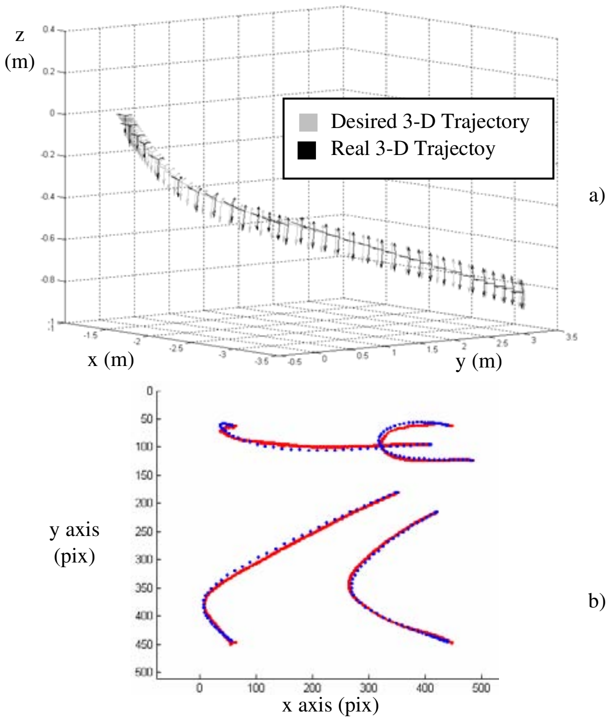


Fig. 7. a) Comparison of the real trajectory and the desired one in the 3-D space. b) Comparison of the sampling of the desired trajectory and the one obtained in the image (Trajectory 1).

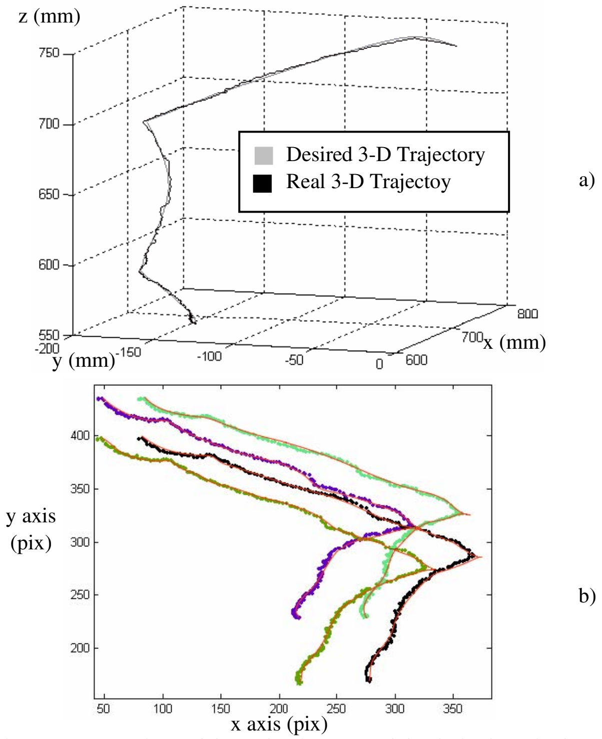


Fig. 8. a) Comparison of the real trajectory and the desired one in the 3-D space. b) Comparison of the sampling of the trajectory obtained and the desired one in the image (Trajectory 2).

C. *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 [5] use a time-variable reference. The time-dependent system tries to maintain the time-references even if the tracking is not carried out correctly, therefore, do not guarantee the correct tracking. 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 (the system for fusing force and visual information shown in our previous paper [12] is used). In such a situation, a time-dependent tracking system based on visual servoing has the behaviour 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 (see Fig. 9.b), once the obstruction has ended, the system continues with the tracking and is not affected by the delay.

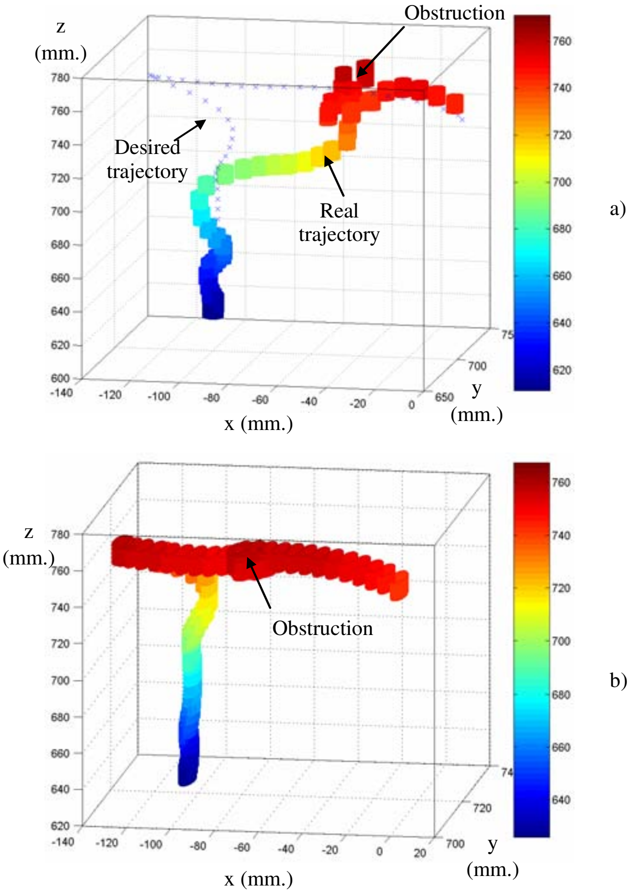


Fig. 9. Trajectory with obstruction. a) 3-D Trajectory obtained using a time-dependent tracking system based on visual servoing. b) 3-D Trajectory obtained by the movement flow-based visual servoing system.

V. Conclusions

The classical image-based visual servoing systems have a limited behaviour when they are applied to tracking trajectories, especially if these trajectories join two distant points. However, their behaviour is quite good when the initial and final features are close, assuring the convergence of the initial features with the final ones. With these considerations, the 2-D movement flow-based visual servoing approach allows us to specify the evolution of the features that will be captured by the camera during the trajectory, while assuring the robustness and convergence of the system.

A fundamental aspect of the strategy described in this paper is that it allows the tracking of non-timed trajectories in the image. As the results obtained show, the use of a 2-D movement flow-based visual servoing allows the tracking of trajectories in the image space, avoiding the limitations and problems that arise with visual servoing systems that are time-dependent, which have been employed up to now for the tracking of trajectories in the image space. From the tests carried out, it can be concluded that with the proposed system a correct tracking of the desired trajectory in the 3-D Cartesian space is obtained.

References

[1] F. Chaumette, “Potential problems of stability and convergence in image-based and position-based visual servoing”, In The Confluence of Vision and Control, vol. 237, D. Kriegman, G. Hager and A. Morse, Eds. Springer Verlag, 1998, pp. 66-78.

[2] S. Hutchinson, G. D. Hager, and P. I. Corke, “A tutorial on visual servo control”, *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 651-670, Oct. 1996.

[3] W. J. Wilson, C. C. Williams, and G. S. Bell. “Relative end-effector control using cartesian position-based visual servoing”, *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 684-696, Oct. 1996.

[4] P. Zanne, G. Morel, and F. Plestan. “Sensor-based Robot Control in the Presence of Uncertainties: Bounding the Task Function Tracking Errors”. IEEE Int. Conf. on Robotics and Automation, ICRA ’02, Washington D.C., May 2002.

[5] Y. Mezouar and F. Chaumette. “Path Planning For Robust Image-based Control”, *IEEE Trans. Robot. Automat.*, vol. 18, no. 4, pp. 534-549, Aug. 2002.

[6] J. Pomares, F. Torres, and S. T. Puente. “Disassembly movements for geometrical objects through heuristic methods”, in *SPIE Int. Symposium on Intelligent Systems and Advanced Manufacturing,* Boston, USA, November 2002, Vol. 4569, pp. 71-80.

[7] S. T. Puente, F. Torres, and J. Pomares. “Product disassembly scheduling using graph models”, in *SPIE Int. Symposium on Intelligent Systems and Advanced Manufacturing*, Boston, USA, November 2002, vol. 4569, pp. 63-70.

[8] J. Serra, Image Analysis and Mathematical Morphology, New York: Academic Press, 1981.

[9] R. Lotufo and F. Zampirolli, “Fast multidimensional parallel euclidean distance transform based on mathematical morphology”, in *Proc. of SIBGRAPI 2001, XIV Brazilian Symposium on Computer Graphics and Image Processing, Florianópolis, Brasil*, October 2001, pp. 100-105.

[10]P. Y. Li and R. Horowitz. “Passive Velocity Field Control (PVFC): Part II – Application to Contour Following”. *IEEE Trans. Automat. Contr.*, vol. 46, no 9. pp. 1360-1371, Sept. 2001.

[11]J. Pomares, F. Torres, and P. Gil. “2-D Visual servoing with integration of multiple predictions of movement based on Kalman filter”, in *15th IFAC World Congress on Automatic Control*, Barcelona, Spain, 2002, vol. L.

[12]J. Pomares, F. Torres, and P. Gil. “Visual servoing and force control fusion for complex insertion tasks”, in *11th International Conference on Advanced Robotics,* Coimbra, Portugal, July 2003, vol. 1, pp. 357-362.