MOVEMENT FLOW-BASED VISUAL SERVOING TO TRACK MOVING OBJECTS

J. POMARES, UNIVERSITY OF ALICANTE, SPAIN, jpomares@ua.es.
F. TORRES, UNIVERSITY OF ALICANTE, SPAIN, Fernando.Torres@ua.es.

ABSTRACT

The purpose of this paper is to describe a new method for tracking trajectories specified in the image space. This method, called movement flow-based visual servoing system, is applied to an eye-in-hand robot and it is shown that it allows the correct tracking of a trajectory, not only in the image but also in the 3-D space. This method is also extended to the case in which the object from which the features are extracted, is in motion. To do so, the estimations obtained, using several Kalman filters, are integrated in the control action.

KEYWORDS: tracking trajectories, image-based visual servoing, movement flow, movement estimation.

1. INTRODUCTION

Using an image-based visual servoing system, a trajectory between the initial and the desired configurations in the image cannot be specified. Only the desired features in the image are indicated and not the 3-D trajectory that should be followed to arrive at such a configuration. In this paper a new approach to track trajectories called movement flow-based visual servoing that presents a correct behaviour not only in the image but also in the 3-D space, is described.

Only recently have appeared studies about the use of visual servoing systems to follow trajectories in the image [1]. In this last paper, an approach that uses an image-based visual servoing system with a time variable reference to track a desired trajectory, is presented. In contrast to the approach described by Mezouar and Chaumette, the one described in this article is used to track non-timed trajectories. If the tracking is formulated as a timed trajectory, the system risks not following the desired trajectory, at the cost of trying to maintain time restrictions.

This paper presents a solution for tracking image trajectories and an eye-in-hand robotic system is applied to confirm the validity of the approach. 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, affording a tracking of the desired trajectory in the image.

The tracking of moving objects is the object of recent researches (see [2]). Within the field of visual servoing it is possible to find several works for real time tracking [3][4]. In this paper the estimation of the movement of the object with respect to which the features are extracted is integrated in the movement flow-based visual servoing system. This aspect allows to track the desired trajectory not only when the object is fixed, but also when is in motion.

To evaluate the trajectory tracking system with and without movement of the object from which the features are extracted, the setup shown in Figure 1.a is employed. The robot that carries out the tracking is a 7 d.o.f Mitsubishi PA-10 with an eye-in-hand camera system composed of a JAI-M536 mini-camera. We have used a 5 d.o.f. SCORBOT ER-IX robot to move the object from which the four features are extracted to evaluate the behaviour of the proposed system.

This paper is organized as follows: firstly the movement flow-based visual servoing system is described. To do so, first the main formulation of the visual servoing system is shown. Then the concept of movement flow and the potential function used is described. Finally, the results obtained and the main conclusions arrived are presented.

2. MOVEMENT FLOW-BASED VISUAL SERVOING

In this section the general concept of movement flow is described, first however, the main formulation of the visual servoing system is detailed.

2.1 Visual servoing

The control actions generated to track a trajectory are obtained from an image-based visual servoing system in which the references are obtained from the movement flow:

$$\boldsymbol{v}^{\mathrm{C}} = -\lambda \cdot \hat{\mathbf{J}}_{\mathrm{f}}^{+} \cdot \boldsymbol{e}_{\mathrm{f}} - \frac{\partial \hat{\boldsymbol{e}}}{\partial t} \qquad (1)$$

where $\boldsymbol{v}^{\mathrm{C}}$ is the velocity obtained from the camera coordinate frame; $\lambda > 0$ is the gain of the controller; $\hat{\mathbf{J}}_{\mathrm{f}}^{+}$ is the pseudoinverse of the interaction matrix [5]; $\boldsymbol{e}_{\mathrm{f}} = \mathbf{s} - \mathbf{s}_{\mathrm{d}}$, $\mathbf{s}=[\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_{\mathrm{M}}]^{\mathrm{T}}$ are the set of selected visual information to develop the task (features extracted from the image); $\mathbf{s}_{\mathrm{d}}=[\boldsymbol{f}_1+m_1\Phi_1(\boldsymbol{f}_1), \boldsymbol{f}_2+m_2\Phi_2(\boldsymbol{f}_2), \ldots, \boldsymbol{f}_{\mathrm{M}}+m_{\mathrm{M}}\Phi_{\mathrm{M}}(\boldsymbol{f}_{\mathrm{M}})]^{\mathrm{T}}$; Φ_{i} is the movement flow for the feature i, $\boldsymbol{m}=\{m_1, m_2, \ldots, m_{\mathrm{M}}\}$ determines the progression speed.

In Equation (1), the term $\partial \hat{\boldsymbol{e}} / \partial t$ represents an estimation of the variations of $\boldsymbol{e} = \hat{\mathbf{J}}_{\mathrm{f}}^{+} \cdot \boldsymbol{e}_{\mathrm{f}}$ due to the movement of the object from which the features are extracted. As is shown in [6], the estimation of the velocity of a moving object tracked with an eye-in-hand camera system can be obtained from the measurements of the camera velocity and from the error function. Thus, from Equation (1) the value of the estimation of the error variation due to the movement of the tracked object can be obtained in this way (to obtain an exponential decrease of the error it must fulfil that $\dot{\boldsymbol{e}} = -\lambda \cdot \boldsymbol{e}$):

$$\frac{\partial \hat{\boldsymbol{e}}}{\partial t} = \dot{\boldsymbol{e}} - \boldsymbol{v}^{\mathrm{C}} \qquad (2)$$

From Equation (2), to acquire the value of the motion estimation in each iteration, it can be obtained using the following expression:

$$\left(\frac{\partial \hat{\boldsymbol{e}}}{\partial t}\right)_{\mathrm{k}} = \frac{\boldsymbol{e}_{\mathrm{k}} - \boldsymbol{e}_{\mathrm{k-1}}}{\Delta t} - \boldsymbol{v}_{\mathrm{k-1}}^{\mathrm{C}} \qquad (3)$$

where Δt can be obtained determining the delay in each iteration of the algorithm, $\boldsymbol{e}_{\mathrm{k}}$ and $\boldsymbol{e}_{\mathrm{k-1}}$ are the error values at the instants k and k-1, and $\boldsymbol{v}_{\mathrm{k-1}}^{\mathrm{C}}$ is the camera velocity measurement at the instant k-1 with respect the camera coordinate frame. This motion estimation must be filtered due to possible imprecision in the camera velocity measurement, $\boldsymbol{v}_{\mathrm{k-1}}^{\mathrm{C}}$, or even in the extraction of the features **s**. To do so, we have used a filter that employs a set of Kalman filters whose formulation is detailed in our previous paper [7].

2.2 Movement flow definition

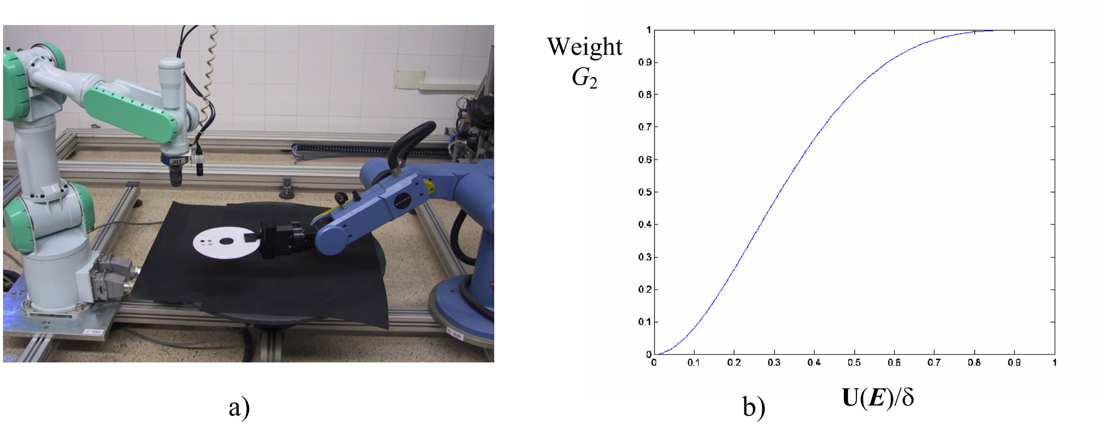
This section describes how to obtain the movement flow for one feature. Therefore, for the sake of clarity, the sub-index that indicates which feature is being considered, is omitted in this section and in the following one.

The movement flow, Φ, is a set of vectors that converge towards the desired image. 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.

To define the movement flow, consider, for a given feature, a desired parameterized trajectory in the image $\boldsymbol{f}_{\mathrm{d}}:\Gamma \rightarrow \Im$ where $\Gamma \subset \Re$. Considering that the coordinates of this trajectory in the image are $\boldsymbol{f}_{\mathrm{d}}(\tau)=[f_{\mathrm{xd}}(\tau), f_{\mathrm{yd}}(\tau)]$ and that $\boldsymbol{f}$ are the current coordinates of the feature in the image, the error vector $\boldsymbol{E}(\boldsymbol{f})=(E_{\mathrm{x}}, E_{\mathrm{y}})$ where $E_{\mathrm{x}}=(f_{\mathrm{x}}-f_{\mathrm{xd}})$ and $E_{\mathrm{y}}=(f_{\mathrm{y}}-f_{\mathrm{yd}})$ is defined, where $\boldsymbol{f}_{\mathrm{d}}=(f_{\mathrm{xd}}, f_{\mathrm{yd}})$ are the coordinates of the nearest point to $\boldsymbol{f}$ in the desired trajectory. Thus, the movement flow, Φ, is defined as a linear combination of two terms:

$$\Phi = G_1\left(\boldsymbol{f}\right)\cdot\begin{pmatrix}\dfrac{\partial f_{\mathrm{xd}}\left(\tau\right)}{\partial\tau}\\ \dfrac{\partial f_{\mathrm{yd}}\left(\tau\right)}{\partial\tau}\end{pmatrix} - G_2\left(\boldsymbol{f}\right)\cdot\begin{pmatrix}\dfrac{\partial\mathbf{U}}{\partial E_{\mathrm{x}}}\\ \dfrac{\partial\mathbf{U}}{\partial E_{\mathrm{y}}}\end{pmatrix} \tag{4}$$

where **U** is the potential function defined in the following section and G_1, G_2: $\Im \to \Re^+$ are weight functions so that $G_1 + G_2 = 1$. The first term in (4), $\dot{\boldsymbol{f}}_d\left(\tau\right)$, is obtained by calculating the Taylor coefficients of the desired trajectory. As can be seen in Equation (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, 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, to determine the values of these weight functions we have used the function shown in Figure 1.b and we have defined the parameter δ being a variable that represents an error value such that if $\mathbf{U}\left(\boldsymbol{E}\left(\boldsymbol{f}\right)\right) > \delta \to G_1 = 0$ (maximum tracking error permitted).



a) b)

Figure 1. a) Experimental setup. b) Evolution of G_2.

As is shown in [8], applying the velocity field codified by Φ, the expected evolution for the error, ***E**,* for $\beta > 0$, will be:

$$\dot{E}_{\mathrm{i}} = -\beta\cdot G_2\left(\boldsymbol{f}\right)\cdot\frac{\partial\mathbf{U}}{\partial E_{\mathrm{i}}} \quad \text{for i = x, y.} \tag{5}$$

Thus, the error evolves in the direction of the negative gradient of the potential and converges at a point of **U** in which $E_{\mathrm{x}}=E_{\mathrm{y}}=0$, so that $\boldsymbol{f}\to\boldsymbol{f}_{\mathrm{d}}$.

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 co-ordinated way, so that the shortest trajectories diminish their velocity to adapt it to the progression of the longest ones. Therefore, in 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 be observed at a desired configuration of the camera in the 3-D Cartesian space. The progression speed, ***m***, of 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.

So far, the systems that use image-based visual servoing to track trajectories [1] employ a time-variable reference. Such systems do not guarantee the correct tracking of the trajectory since the references can be very restrictive and, therefore, the system tries to maintain the time-references, even if the tracking is not performed correctly. This problem is solved by using movement flow-based visual servoing, as this system is not affected by time restrictions (i.e., the references are obtained from the movement flow and not from a time-dependent function). In an experiment in which the robot must interact with objects within the workspace, it may be obstructed for a certain time. In such a situation, if a time-dependent tracking system based on visual servoing is employed, the references are delayed and, therefore, are very restrictive and do not afford the correct tracking. However, using movement flow-based visual servoing, once the obstruction has ended, the system continues with the tracking and is not affected by the delay.

2.3 Potential Function

The potential function must attain its minimum when the error is zero and must increase as $\boldsymbol{f}$ deviates more from the desired location $\boldsymbol{f}_\mathrm{d}$. $\mathbf{I}$ is 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 $\mathbf{I}_\mathrm{g}$ of $\mathbf{I}$ [9]. Once the image $\mathbf{I}_\mathrm{g}$ has been obtained, the next step to determine the potential function is to generate a distance map [10]. The distance map creates a distance image $\mathbf{I}_\mathrm{d}$ of the image $\mathbf{I}_\mathrm{g}$, so that the value of $\mathbf{I}_\mathrm{d}$ at the pixel x is the Euclidean distance of x from the complement of $\mathbf{I}_\mathrm{g}$. In Figure 2.a, a three-dimensional representation of the distance map is shown for the feature $\boldsymbol{f}_1$ used in the results section (see Figure 3.a). In this figure the value of z coordinate represents the distance between each pixel and the nearest pixel to it in the desired trajectory. This representation shows the distribution of the potential function.

Using this potential function and following the steps described, the movement flow can be obtained whatever the trajectory might be. In Figure 2.b a detail of the movement flow obtained for the trajectory whose distance map is represented in Figure 2.a, is shown.

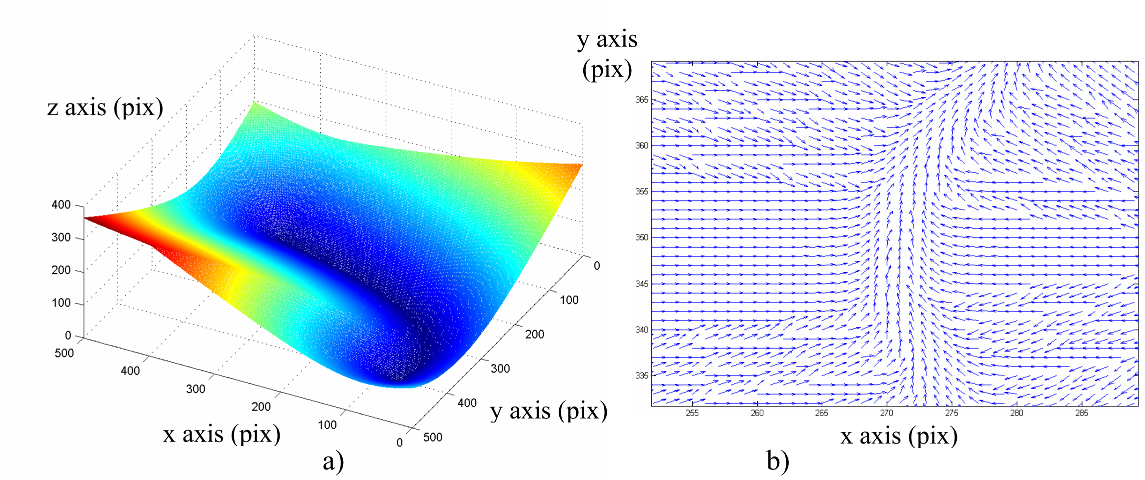


Figure 2. a) Distance map. b) Detail of the movement flow obtained.

3. RESULTS

3.1 Tracking Trajectories

First, in this section, the behaviour of the movement flow-based visual servoing system for tracking trajectories when the object is fixed is shown. Figure 3.a and 3.b show the desired trajectory that must be followed by the camera in the image and in the 3-D space respectively. Once the movement flow has been determined (δ = 10), and considering the concepts of visual servoing shown in this paper, the trajectories of the features in the image, represented in Fig. 3.a, are obtained (Figure 3.b shows the trajectory of the robot end-effector). The correct behaviour of the system, not only in the image but also in the 3-D space, is observed.

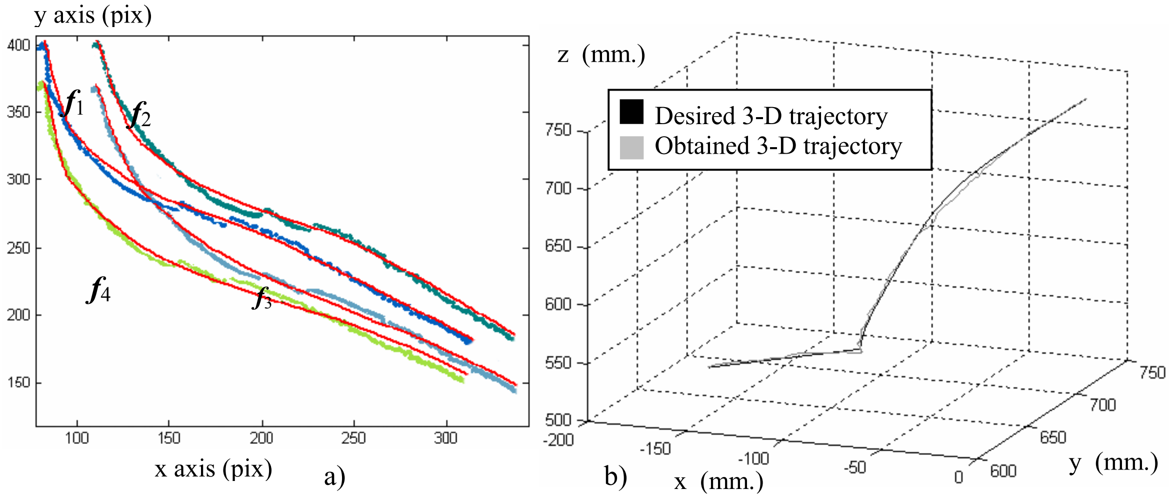


Figure 3. a) Comparison of the sampling of the obtained trajectory and the one desired in the image. b) Comparison of the obtained trajectory and the desired one in the 3-D space.

3.2 Tracking Moving Objects

In this case, the SCORBOT ER-IX robot moves the object from which the features are extracted. In Figure 4.a the trajectory described by the system in the image space with and without considering motion estimation, is represented. The desired trajectory is also shown. It can be observed that when we introduce the estimation in the system, the error caused by the movement of the object is considerably reduced. Thus, without estimation the system is able to arrive to the final of the trajectory when the object motion ends. This is because when the tracking reaches a point in the image (marked with arrows in Figure 4), the movement flow tries to reduce the error caused by the object motion, however, this control action cannot compensate the motion, and only can reduce the error when the motion ends.

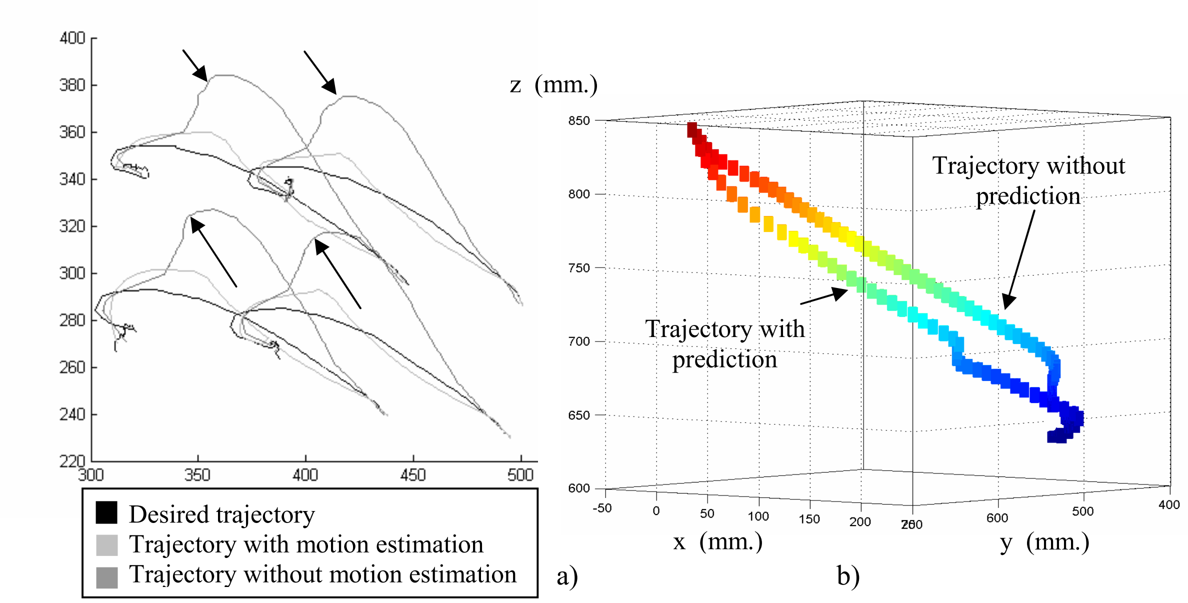


Figure 4. a) Comparison between the desired trajectory in the image and the one obtained with and without motion estimation. b) Comparison between the 3-D trajectories obtained with and without motion estimation.

In Figure 4.b the difference between the 3-D trajectory obtained with and without motion estimation is analyzed. It can be observed that integrating the estimation in the control action, the system continues with the tracking of the trajectory although the object was in motion. It can also be observed that if the system does not includes the motion estimation only when the motion ends, the system is able to continue with the tracking. Regarding the processing time required to compute the movement flow, we should mention that is calculated progressively as the tracking is done, obtaining an average delay in each iteration of 1,349msec. This value is negligible compared with the time required for the capture and processing of the image (40 msec).

4. CONCLUSIONS

Up to now, the majority of the image-based visual servoing systems allow to achieve a desired configuration of the features in the image from the initials one. However, it is not possible to know the trajectory that el robot follows between such configurations. For this purpose, a new approach to track trajectories specified with respect to moving objects called movement flow-based visual servoing that includes an estimation of the error due to the movement of the tracked object, is presented. The systems that use image-based visual servoing to track trajectories that have been employed up to now 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. The results demonstrate the correct tracking of the desired trajectory when the object from which the features are extracted is in motion. The integration of the proposed estimator in the movement flow compensates the object motion and obtains the correct tracking (in the image and in the 3-D space).

5. REFERENCES

[1] Y. Mezouar and F. Chaumette, "Path Planning For Robust Image-based Control," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 4, 2002, pp. 534-549.

[2] D. Comaniciu, V. Armes and P. Meer, "Kernel-Based Object Tracking," *IEEE Transactions on Patter analysis and machine intelligence*, Vol. 25, No. 5, 2003, pp. 564-576.

[3] T. Drummond and R. Cipolla, "Real-Time Visual Tracking of Complex Structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, 2002, pp. 932-946.

[4] J. P. Barreto, J. Batista and H. Araujo, "Model Predictive Control to Improve Visual Control of Motion: Applications in Active Tracking of Moving Targets," Proc. of the International Conference on Pattern Recognition, Barcelona, Spain 2000, pp. 4732-4736.

[5] S. Hutchinson, G. D. Hager and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, 1996, pp. 651-670.

[6] F. Bensalah and F. Chaumette, "Compensation of abrupt motion changes in target tracking by visual servoing," Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1, Pittsburgh, Pennsylvania, USA 1995, pp. 181-187.

[7] J. Pomares, F. Torres and P. Gil, "2-D Visual servoing with integration of multiple predictions of movement based on Kalman filter," Proc. of IFAC 2002 15th World Congress, Vol L, Barcelona, Spain 2002.

[8] P. Y. Li and R. Horowitz, "Passive Velocity Field Control Approach to Robot Contour Following," Proc of the Japan USA Symposium on Flexible Automation, Vol. 1, Kobe, Japan 1996, pp. 25-32.

[9] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, London, 1981.

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