

# IMPROVING TRACKING TRAJECTORIES WITH MOTION ESTIMATION

J. Pomares, G. J. García, F. Torres

*Physics, Systems Engineering and Signal Theory Department. University of Alicante, Alicante, Spain*

*Email: {jpomares, gjgg, Fernando.Torres}@ua.es*

Keywords: Motion estimation, tracking trajectories, visual control.

Abstract: Up to now, different methods have been proposed to track trajectories using visual servoing systems. However, when these approaches are employed to track trajectories specified with respect to moving objects, different considerations must be included in the visual servoing formulation to progressively decrease the tracking error. This paper shows the main properties of a non-time dependent visual servoing system to track image trajectories. The control action obtained integrates the motion estimation of the object from which the features are extracted. The proposed motion estimator employs information from the measures of the extracted features and from the variation of the camera locations. These variations are obtained determining the Homography matrix between consecutive camera frames.

## 1. INTRODUCTION

Now, visual servoing systems are a well-known approach to guide a robot using image information. Typically, visual servoing systems are position-based and image-based classified (Hutchinson et al. 1996). Position-based visual servoing requires the computation of a 3-D Cartesian error for which a perfect CAD-model of the object and a calibrated camera are necessary. These types of systems are very sensitive to modeling errors and noise perturbations. In image-based visual servoing the error is directly measured in the image. This approach ensures the robustness with respect to modeling errors, but generally an inadequate 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. This nice property ensures a correct convergence if the desired configuration is sufficiently near to the current one. This paper shows the properties of an approach which employs image-based visual servoing to track trajectories. This method is not the main objective of the paper and a more extensive study of the approach can be seen in our previous works (Pomares and Torres, 2005).

Once this strategy is defined, the paper focuses on the extension of the previous mentioned algorithms

to be able to carry out the tracking of image trajectories when the visual features are in motion. Previous works, such as (Hutchinson et al. 1996) have shown the necessity of estimating object motion and to include this estimation in the control action in order to decrease the tracking errors. The motion estimation can be solved using different algorithms like the one shown in (Pressigout and Marchand, 2004) based in virtual visual servoing. In (Bensalah and Chaumette, 1995) an estimator which employs measurements about the camera velocity is proposed. However, this method introduces errors in the estimation due to the non accuracy measurement of the camera motion. The previous mentioned algorithms are used to achieve a given configuration of the features in the image. In this paper, we define method to improve the estimation of the camera velocity based on visual information. This method is used to define a motion estimator to be applied during the tracking of trajectories.

This paper is organized as follows: The main aspects of the visual servoing system to track trajectories are first described in Section 2. Section 3 shows a method to estimate the motion of the object from which the features are extracted. Section 4, simulation and experimental results confirm the validity of the proposed algorithms. The final section presents the main conclusions arrived at.

## 2. TRACKING IMAGE TRAJECTORIES

First, the notation employed and the desired trajectory to be tracked are described. In this paper, we suppose that the robot must track a desired trajectory in the 3D space,  $\gamma(t)$ , using an eye-in-hand camera system. By sampling the desired trajectory,  $\gamma(t)$ , a sequence of  $N$  discrete values is obtained, each of which represents  $N$  intermediate positions of the camera  ${}^k\gamma/k \in 1 \dots N$ . From this sequence, the discrete trajectory of the object in the image  $S = \{ {}^k\mathbf{s}/k \in 1 \dots N \}$  can be obtained, where  ${}^k\mathbf{s}$  is the set of  $M$  points or features observed by the camera at instant  $k$ ,  ${}^k\mathbf{s} = \{ {}^k\mathbf{f}_i/i \in 1 \dots M \}$ . In the next section, a non time-dependent visual controller to track the previous mentioned image trajectory is described.

### 2.1 Visual servo control

The control action obtained from the visual controller is:

$$\mathbf{v}_V^C = -\lambda \cdot \hat{\mathbf{J}}_f^+ \cdot \mathbf{e}_f \quad (1)$$

where  $\mathbf{v}_V^C$  is the velocity obtained with respect to the camera coordinate frame;  $\lambda > 0$  is the gain of the controller;  $\hat{\mathbf{J}}_f^+$  is the pseudoinverse of the estimated interaction matrix (Hutchinson et al. 1996);  $\mathbf{e}_f = \mathbf{s} - \mathbf{s}_d$ ;  $\mathbf{s} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_M]^T$  are the set of features extracted from the image;  $\mathbf{s}_d = [\mathbf{f}_1 + m_1\Phi_1(\mathbf{f}_1), \mathbf{f}_2 + m_2\Phi_2(\mathbf{f}_2), \dots, \mathbf{f}_M + m_M\Phi_M(\mathbf{f}_M)]^T$ ;  $\Phi_i$  is the movement flow for the feature  $i$ ,  $\mathbf{m} = \{m_1, m_2, \dots, m_M\}$  determines the progression speed.

Now, for the sake of clarity, the sub-index that indicates which feature is being considered and the super-index that indicates the instant in which these features are obtained are omitted. The movement flow,  $\Phi$ , is a set of vectors converging towards the desired trajectory in the image.

We consider, for a given feature, a desired parameterized trajectory in the image  $\mathbf{f}_d: \Gamma \rightarrow \mathfrak{S}$  where  $\Gamma \subset \mathfrak{R}$ . The coordinates of this trajectory in the image are  $\mathbf{f}_d(\tau) = [f_{xd}(\tau), f_{yd}(\tau)]$  and  $\mathbf{f}$  are the current coordinates of the feature in the image. The error vector  $\mathbf{E}(\mathbf{f}) = (E_x, E_y)$  where  $E_x = (f_x - f_{xd})$  and  $E_y = (f_y - f_{yd})$  is defined, where  $\mathbf{f}_d = (f_{xd}, f_{yd})$  are the coordinates of the nearest point to  $\mathbf{f}$  in the desired trajectory. The movement flow,  $\Phi$ , is defined as:

$$\Phi = G_1(\mathbf{f}) \cdot \begin{pmatrix} \frac{\partial f_{xd}(\tau)}{\partial \tau} \\ \frac{\partial f_{yd}(\tau)}{\partial \tau} \end{pmatrix} - G_2(\mathbf{f}) \cdot \begin{pmatrix} \frac{\partial U}{\partial E_x} \\ \frac{\partial U}{\partial E_y} \end{pmatrix} \quad (2)$$

where  $U$  is the potential function defined in Section 3.2 and  $G_1, G_2: \mathfrak{S} \rightarrow \mathfrak{R}^+$  are weight functions so that  $G_1 + G_2 = 1$ . As can be seen in Equation (2), 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. Specifically, to determine the values of these weight functions we have used the function shown in Figure 1 and we have defined the parameter  $\delta$  being a variable that represents an error value such that if  $U(\mathbf{E}(\mathbf{f})) > \delta \rightarrow G_1 = 0$  (maximum tracking error permitted).

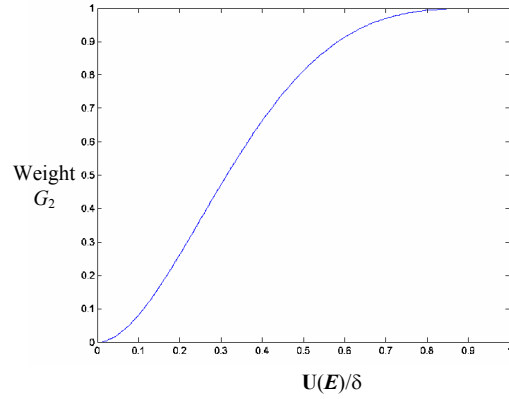


Figure 1: Evolution of  $G_2$ .

### 2.2 Potential Function

The potential function  $U$  employed in the definition of the movement flow must attain its minimum when the error is zero and must increase as  $\mathbf{f}$  deviates more from the desired location  $\mathbf{f}_d$ . The visual features are under the influence of an artificial potential field ( $U$ ) defined as an attractive potential pulling the features towards the desired image trajectory.  $\mathbf{I}$  is the image that would be obtained after the trajectory  $\mathbf{f}_d(\tau)$  has been represented. The first step in determining the potential function is to calculate the gradient  $\mathbf{I}_g$  of  $\mathbf{I}$ . Once the image  $\mathbf{I}_g$  has been obtained, the next step to determine the potential function is to generate a distance map (Lotufo and Zampiroli, 2001). The distance map

creates a distance image  $\mathbf{I}_d$  of the image  $\mathbf{I}_g$ , so that the value of  $\mathbf{I}_d$  at the pixel  $x$  is the Euclidean distance from  $x$  to the complement of  $\mathbf{I}_g$ . In Figure 2, a three-dimensional representation of the distance map is shown for a given feature. 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,  $\mathbf{U}$ . Using this potential function and following the steps described, the movement flow can be obtained. In Figure 3 a detail of the movement flow obtained for the trajectory whose distance map is represented in Figure 2, is shown.

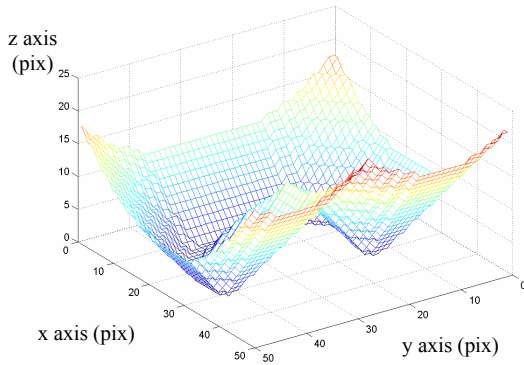


Figure 2. Distance map.

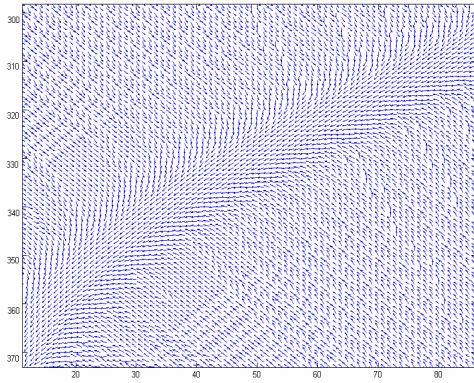


Figure 3: Movement flow.

### 3. TRACKING MOVING OBJECTS

In order to track trajectories specified with respect to moving objects, the motion of the object must be included in the control action proposed in (1). Doing so, the new control action will be:

$$\mathbf{v}^c = -\lambda \cdot \hat{\mathbf{J}}_f^+ \cdot \mathbf{e}_f - \frac{\partial \hat{\mathbf{e}}}{\partial t} \quad (3)$$

where  $\frac{\partial \hat{\mathbf{e}}}{\partial t}$  represents the estimation of the variations of  $\mathbf{e} = \hat{\mathbf{J}}_f^+ \cdot \mathbf{e}_f$  due to the movement of the object from which the features are extracted. As is shown in (Bensalah and Chaumette, 1995), 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  $\dot{\mathbf{e}} = -\lambda \cdot \mathbf{e}$ ):

$$\frac{\partial \hat{\mathbf{e}}}{\partial t} = \dot{\mathbf{e}} - \mathbf{v}^c \quad (4)$$

From Equation (4), the value of the motion estimation can be obtained using the following expression:

$$\left( \frac{\partial \hat{\mathbf{e}}}{\partial t} \right)_k = \frac{\mathbf{e}_k - \mathbf{e}_{k-1}}{\Delta t} - \mathbf{v}_{k-1}^c \quad (5)$$

Where  $\Delta t$  can be obtained determining the delay at each iteration of the algorithm,  $\mathbf{e}_k$  and  $\mathbf{e}_{k-1}$  are the error values at the instants  $k$  and  $k-1$ , and  $\mathbf{v}_{k-1}^c$  is the camera velocity measurement at the instant  $k-1$  with respect the camera coordinate frame. As is shown in our previous works (Pomares et al. 2002) the estimations obtained from Equation (5) depends on the measurement of the camera motion which cannot be measured without errors. Therefore, in order to improve the global behaviour of the system it is necessary to obtain a more accuracy estimation of the camera motion.

To compute the camera velocity in the previous iteration  $\mathbf{v}_{k-1}^c$  it is necessary to obtain the rotation  $\mathbf{R}_{k-1}$  and translation  $\mathbf{t}_{dk-1}$  between the two last frames. To do so, first of all, we call  $\Pi$  the plane containing the object. Considering  $P$  a 3D point observed by the camera, the same point in the image space is  $p$ . With  $p_{k-1}$  we represent the position of the feature in the image captured at instant  $k-1$ . In the next image (which corresponds with the image obtained by the camera after one iteration of the control loop), the same point will be located at  $p_k$  position (see Figure 4). The projective homography  $\mathbf{G}$  is defined as:

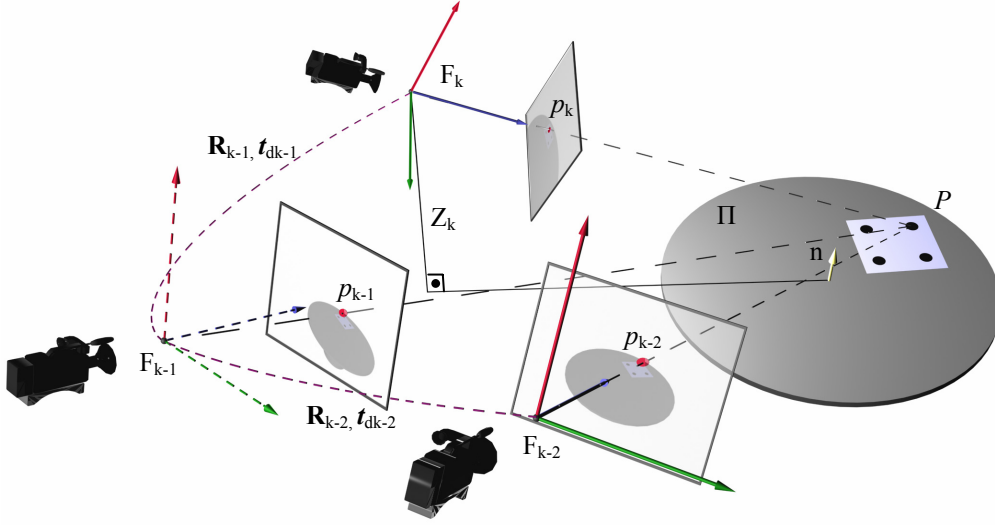


Figure 4: Scheme of the motion estimation.

$$\mu_{k-1} p_{k-1} = \mathbf{G} p_k + \beta \zeta \quad (6)$$

where  $\mu_{k-1}$  is a scale factor,  $\zeta = \mathbf{A} \mathbf{R}_{k-1} \mathbf{t}_{dk-1}$  is the projection in the image captured by the camera at time  $k-1$  of the optical centre when the camera is in the next position, and  $\beta$  is a constant scale factor which depends on the distance  $Z_k$  from the contact surface to the origin of the camera placed at the current position:

$$\beta = \frac{d(P, \Pi)}{Z_k} \quad (7)$$

where  $d(P, \Pi)$  is the distance from the contact surface plane to the 3D point. Although  $\beta$  is unknown, applying (6) between the instant  $k-1$  (at previous iteration) and the current camera positions we can obtain that:

$$\beta = \text{sign} \left( \frac{(\mu_{k-1} p_{k-1} - \mathbf{G}_{k-1} p_k)_1}{(\mathbf{A} \mathbf{R}_{k-1} \mathbf{t}_{dk-1})_1} \right) \frac{\|\mathbf{G}_{k-1} p_k \wedge p_{k-1}\|}{\|\mathbf{A} \mathbf{R}_{k-1} \mathbf{t}_{dk-1} \wedge p_{k-1}\|} \quad (8)$$

where subscript 1 indicates the first element of the vector and  $\mathbf{A}$  is a non singular matrix containing the camera internal parameters:

$$\mathbf{A} = \begin{bmatrix} f \cdot p_u & -f \cdot p_u \cdot \cot(\theta) & u_0 \\ 0 & f \cdot p_v / \sin(\theta) & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

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

If  $P$  is on the plane  $\Pi$ ,  $\beta$  is null. Therefore, from (6):

$$\mu_{k-1} p_{k-1} = \mathbf{G} p_k \quad (10)$$

Projective homography  $\mathbf{G}$  can be obtained through expression (10) if at least four points on the surface  $\Pi$  are given (Hartley and Zisserman, 2000). This way, we can compute projective homography relating previous and current positions  $\mathbf{G}_{k-1}$ . In order to obtain  $\mathbf{R}_{k-1}$  and  $\mathbf{t}_{dk-1}$  we must introduce the concept of the Euclidean homography matrix  $\mathbf{H}$ .

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

$$\mathbf{H} = \mathbf{A}^{-1} \mathbf{G} \mathbf{A} \quad (11)$$

From  $\mathbf{H}$  it is possible to determine the camera motion applying the algorithm shown in (Zhang and Hanson 1996). So, applying this algorithm between previous and current positions we can compute  $\mathbf{R}_{k-1}$  and  $\mathbf{t}_{dk-1}$ .

As iteration time  $\Delta t$  is known, it is now easy to compute the velocity of the camera at previous

iteration  $\mathbf{v}_{k-1}$  through  $\mathbf{R}_{k-1}$  and  $\mathbf{t}_{dk-1}$ . The linear component of the velocity is computed directly from  $\mathbf{t}_{dk-1}$ , whereas the computation of angular velocity requires taking other previous steps. The angular velocity can be expressed as:

$$\boldsymbol{\omega}_{k-1} = \mathbf{T}_{k-1}(\boldsymbol{\varphi}_{k-1})\dot{\boldsymbol{\varphi}}_{k-1} \quad (12)$$

where  $\boldsymbol{\varphi}_{k-1} = [\alpha_{k-1} \ \beta_{k-1} \ \gamma_{k-1}]$  are the Euler angles ZYZ obtained from  $\mathbf{R}_{k-1}$ ,  $\dot{\boldsymbol{\varphi}}_{k-1}$  are the time derivative of the previous Euler angles and:

$$\mathbf{T}_{k-1}(\boldsymbol{\varphi}_{k-1}) = \begin{pmatrix} 0 & -\sin \alpha_{k-1} & \cos \alpha_{k-1} \sin \beta_{k-1} \\ 0 & \cos \alpha_{k-1} & \sin \alpha_{k-1} \sin \beta_{k-1} \\ 1 & 0 & \cos \beta_{k-1} \end{pmatrix} \quad (13)$$

Applying (7) it is possible to compute the angular velocity of the camera to achieve current position from the previous iteration. This way linear and angular velocity of the camera between the two iterations are computed:

$$\mathbf{v}_{k-1} = \begin{pmatrix} \mathbf{v}_{tk-1} \\ \boldsymbol{\omega}_{k-1} \end{pmatrix} = \begin{pmatrix} \frac{\mathbf{t}_{dk-1}}{\Delta t} \\ \mathbf{T}_{k-1}(\boldsymbol{\varphi}_{k-1})\dot{\boldsymbol{\varphi}}_{k-1} \end{pmatrix} \quad (14)$$

## 4. RESULTS

### 4.1 Experimental setup

The system architecture is composed of an eye-in-hand PHOTONFOCUS MV-D752-160-CL-8 camera at the end-effector of a 7 d.o.f. Mitsubishi PA-10 robot. The camera is able to acquire and to process up to 100 frames/second using an image resolution of 320x240. In this paper we are not interested in image processing issues; therefore, the image trajectory is generated using four grey marks whose centres of gravity will be the extracted features.

### 4.2 Simulation results

In this section simulation results are obtained from the application of the proposed visual servoing system to track trajectories specified with respect to moving objects. To do so, the motion estimator

described in Section 3 is used. The motion of the features extracted during the experiment is shown in Figure 5. In this figure the evolution of the visual features observed from the initial camera position is represented (in this figure we consider that any control action is applied to the robot). We can see that the target object describes a periodical and rectangular motion. To better show the object motion, in Figure 6 the image error evolution is represented. As previously, in Figure 6 any control action is applied to the robot (in this figure the desired features are the ones provided by the movement flow). This motion will not end until the trajectory is completely tracked. Figure 7 shows the desired image trajectory and the ones obtained considering and without considering motion estimation. This figure shows that, using motion estimation, the system tracks the desired trajectory avoiding error due to the motion of the object from which the features are extracted.

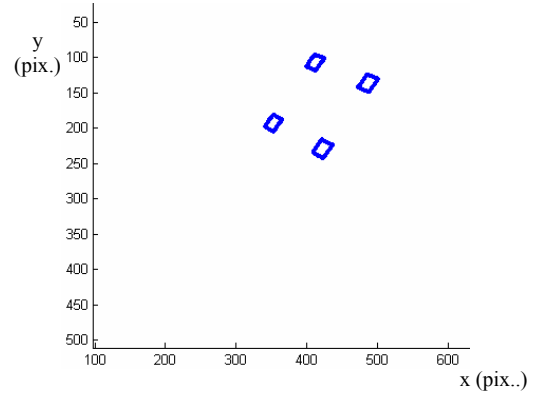


Figure 5: Image trajectory observed from the initial camera position.

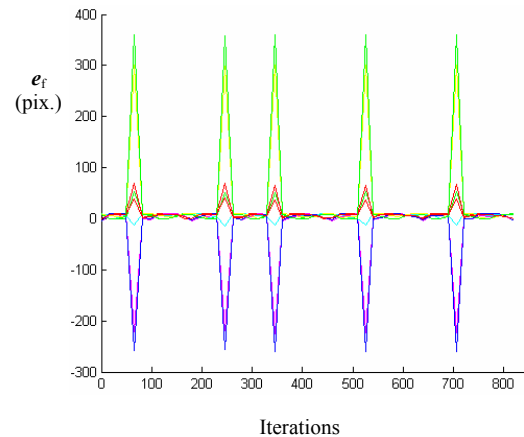


Figure 6: Image error due to the object motion.

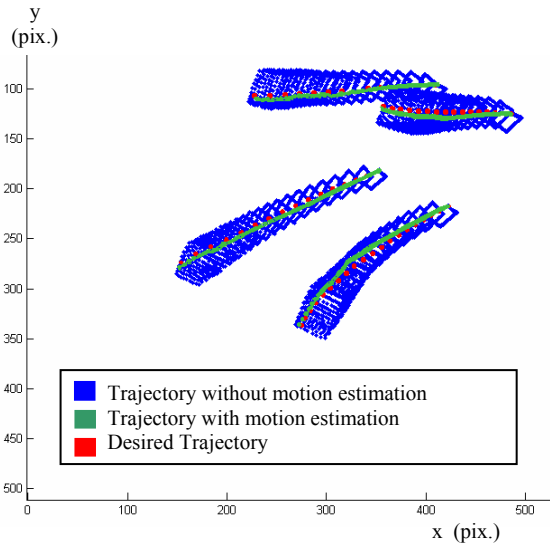


Figure 7: Desired image trajectory and the ones obtained with and without motion estimation.

### 4.3 Experimental results

In order to verify the correct behaviour of the system the experimental setup described in Section 4.1 is applied to track trajectories using the proposed visual servoing system. In Figure 8 the desired trajectory (without considering motion of the object from which the trajectory is specified) is shown. In the same figure is also represented the obtained trajectory when the object from which the features are extracted is in motion. We can observe that the motion is lineal (is not equal to the one described in section 4.2).

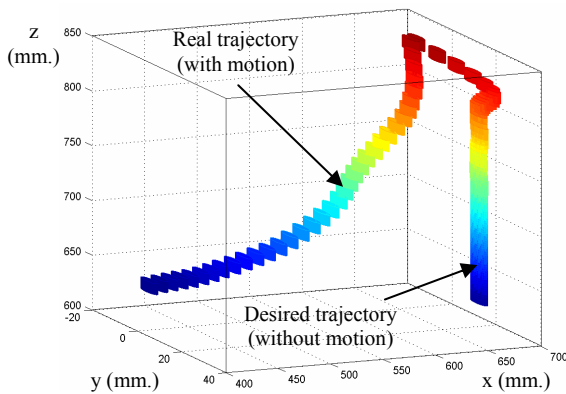


Figure 8: Comparison between the desired 3D trajectory and the one obtained using motion estimation.

In order to observe the improvement introduced by the motion estimation in the visual servoing system, in Figure 9 a comparison in the image space between the desired trajectory and the ones obtained with and without motion estimation is represented. We can see that using motion estimation the system reduces rapidly the tracking error due to the motion of the object. The object motion is not constant, therefore, when the motion estimation is not carried out, the error increases quickly when the object velocity increases. This tracking error is clearly reduced using the motion estimation proposed in this paper.

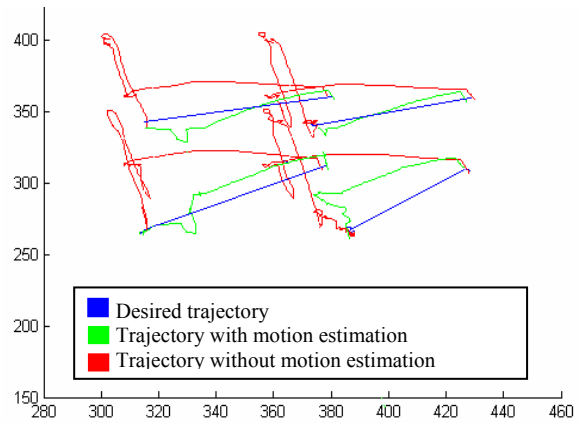


Figure 9: Desired image trajectory and the ones obtained with and without motion estimation.

## 5. CONCLUSIONS

In this paper we have shown a visual servoing system to track image trajectories specified with respect to moving objects. In order to avoid the errors introduced by the motion, it is required to include in the control action of the visual servoing system the effect of this motion. The paper describes the necessity to obtain a good estimation of the camera motion and, to do so, a method based on visual information has been proposed. Simulation and experimental results show that using the proposed estimator a good tracking is obtained avoiding the errors introduced by the motion.

## 6. ACKNOWLEDGEMENTS

This work was funded by the Spanish MCYT project DPI2005-06222 "Diseño, implementación y experimentación de escenarios de manipulación

inteligentes para aplicaciones de ensamblado y desensamblado automático” and by the project GV05/007: “Diseño y experimentación de estrategias de control visual-fuerza para sistemas flexibles de manipulación”.

## 7. REFERENCES

- Bensalah, F., Chaumette, F. 1995. Compensation of abrupt motion changes in target tracking by visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'95*, Pittsburgh, Vol. 1, pp. 181-187.
- Hartley R., Zisserman A. 2000. *Multiple view Geometry in Computer Vision*. Cambridge Univ. Press, pp. 91- 92.
- Hutchinson, S., Hager, G., Corke, P., 1996. A Tutorial on Visual Servo Control. *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651-670.
- Lotufo, R. and Zampiroli, F. 2001. Fast multidimensional parallel euclidean distance transform based on mathematical morphology. In *Proc. of SIBGRAPI 2001*, pp. 100-105.
- Pomares, J., Torres, F., Gil, P. 2002. 2-D Visual servoing with integration of multiple predictions of movement based on Kalman filter. In *Proc. of IFAC 2002 15<sup>th</sup> World Congress*, Barcelona, Vol L.
- Pomares, J., Torres, F., 2005. Movement-flow based visual servoing and force control fusion for manipulation tasks in unstructured environments. *IEEE Transactions on Systems, Man, and Cybernetics—Part C*. Vol. 35, No. 1. Pp. 4 – 15.
- Pressigout, M., Marchand, E. 2004. Model-free augmented reality by virtual visual servoing. In *IAPR Int. Conf. on Pattern Recognition, ICPR'04*, Cambridge.
- Zhang, Z., Hanson, A.R. 1996. Three-dimensional reconstruction under varying constraints on camera geometry for robotic navigation scenarios, Ph.D. Dissertation.