A New Time-Independent Image Path Tracker to Guide Robots Using Visual Servoing

G. J. García, J. Pomares, F. Torres
Physics, Systems Engineering and Signal Theory Department
University of Alicante
PO Box 99, 03080, Alicante. Spain.
{gjgg, jpomares, Fernando.Torres}@ua.es

Abstract

In this paper, a new method to track image trajectories by visual servoing is proposed. This method solves the problem of the previous proposed time-independent tracking systems based on visual servoing. With the proposed method, the robot can track a previously generated trajectory affording a correct tracking not only in the image but also in the 3D space. This new method presents several improvements over the previous ones such as the possibility of specifying the desired tracking velocity, a less oscillating behavior or a correct tracking in the 3D space when high velocities are used. In order to demonstrate the correct behavior of the visual servoing system, an eye-in-hand camera system is used.

1. Introduction

The image-based visual servoing systems have been demonstrated to be robust and efficient to guide a robot towards positions which are near to the initial ones [3]. In these local conditions the system's behaviour is adequate not also in the image space, but also in the 3D Cartesian space. In these systems, the lack of knowledge of the correct camera internal parameters is not crucial for the proper implementation of the task. However, when the distance between the initial and the final position is increased, the effects of the errors in camera calibration parameters are clearly observed [1]. When the camera calibration parameters are not correctly known, a non-linear evolution of the features in the image space is obtained and some image features can disappear.

This problem can be solved developing a visual servoing system which carries out the tracking of trajectories in the image space [7][11]. To do so, a time dependent reference can be used in order to provide the visual servoing systems with the adequate references to track the previously defined trajectory. In [11] the visibility problem is solved allowing some deviation from the desired path to be tracked. The performance of these methods has been demonstrated to be correct even when the correct camera intrinsic parameters are unknown.

When a robot interacts with its environment, most of the methods to track image trajectories proposed up to now fail. These methods are time-dependent. Hence, if the robot collides with an obstacle the trajectory will not be correctly tracked. During the obstruction, the controller continues sending the next timed references. Once the obstruction ends, the robot aims to maintain the time restrictions without allowing the correct tracking of the trajectory.

The method proposed in this paper has a time-independent behaviour. In [10] a time-independent method to track trajectories called movement flow-based visual servoing is presented. This system solves the time-variable reference problem previously addressed. However, the movement flow-based visual servoing has several problems which must be solved in order to be applied in a real workspace. One of these problems is that the visual features are considered individually. It would be a convenient method to track the trajectory of a single feature in the image space. Nevertheless, to achieve a correct visual servoing it is necessary to have at least four features [2]. As it is demonstrated in the present paper, this method does not guarantee a correct tracking in the 3D space when high velocities are employed. Furthermore, it is not possible to specify the desired velocity to develop the tracking. To solve these problems, in this paper, a new time-independent method to track trajectories using visual servoing is presented. This method guarantees the correct tracking in the 3D space.

Over the last few decades, camera calibration methods have been widely investigated (see e.g. [13][14]). Within these methods, virtual visual servoing systems are subject of recent researches [4][5]. Using this approach the camera calibration parameters can be iteratively estimated. In the proposed method, virtual visual servoing is executed at each iteration of the visual servoing task in order to determine the camera location.

This paper is organized as follows: section 2 recalls the basic virtual visual servoing approach; section 3

1-4244-0826-1/07/$20.00 © 2007 IEEE

describes the proposed visual servoing scheme to track trajectories; in section 4 experimental results, using an eye-in-hand system, confirm the validity of the tracking algorithm; the final section presents the main conclusions.

2. Visual Servoing and Virtual Visual Servoing

This section describes the virtual visual servoing approach employed to determine the extrinsic camera parameters during the tracking of the trajectory. Firstly, to present this approach, the basic image-based visual servoing scheme is presented. A visual servoing task can be described by an image function, $\mathbf{e}_t$, which must be regulated to 0:

$$\mathbf{e}_t = \mathbf{s} - \mathbf{s}^* \tag{1}$$

where $\mathbf{s}$ is a M x 1 vector containing M visual features corresponding to the current state, while $\mathbf{s}^*$ denotes the visual features values in the desired state. By imposing an exponential decrease of $\mathbf{e}_t$ ($\dot{\mathbf{e}}_t = -\lambda_1 \mathbf{e}_t$) it is possible to obtain the following control action for a classical image-based visual servoing:

$$\mathbf{v}_c = -\lambda_1 \hat{\mathbf{L}}_s^+ \left(\mathbf{s} - \mathbf{s}^*\right) \tag{2}$$

where $\hat{\mathbf{L}}_S^+$ is the pseudoinverse of an approximation of the interaction matrix [3].

The virtual visual servoing approach is based on the previous mentioned image-based visual servoing scheme. In order to define the virtual visual servoing systems, we consider that the observed features in the image are denoted by $\mathbf{p}_d$ and $\mathbf{p}$ are the current positions of the image features projected using the camera intrinsic parameters, ξ, and the current extrinsic parameters, ${}^c\mathbf{M}_o$ (pose of the object frame with respect to the camera frame).

In order to determine the camera extrinsic parameters, it is necessary to minimize iteratively the error, $\mathbf{e}$, between the observed data, $\mathbf{p}_d$, and the position of the same features $\mathbf{p}$ computed using the following equation:

$$\mathbf{p} = pr_\xi\left({}^c\mathbf{M}_o\, {}^o\mathbf{P}\right) \tag{3}$$

Where ${}^o\mathbf{P}$ are the 3D coordinates (with respect to the object coordinate frame) of the points extracted by the camera and pr_ξ denotes the perspective projection model according to the intrinsic parameters. From the previous defined error, $\mathbf{e}$, it is possible to obtain (the camera intrinsic parameters do not vary):

$$\dot{\mathbf{e}} = \dot{\mathbf{p}} - \dot{\mathbf{p}}_d = \frac{\partial \mathbf{p}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} \tag{4}$$

where $\mathbf{r}$ is the camera pose. Equation (4) can be rewritten as:

$$\dot{\mathbf{e}} = \mathbf{L}_p \mathbf{v} \tag{5}$$

where $\mathbf{v}$ is the instantaneous virtual camera velocity and $\mathbf{L}_p$ the interaction matrix related to $\mathbf{p}$ [6].

To make $\mathbf{e}$ decrease exponentially to 0 ($\dot{\mathbf{e}} = -\lambda_1 \mathbf{e}$) the following control law is obtained:

$$\mathbf{v} = -\lambda_1 \mathbf{L}_p^+ \mathbf{e} \tag{6}$$

Once $\mathbf{e}$ = 0 the extrinsic camera parameters are obtained. In order to observe the behaviour of the virtual visual servoing approach, an experiment to determine the extrinsic camera parameters is presented. The intrinsic parameters considered in this experiment are (u_0, v_0) = (154, 99) and (f_u, f_v) = (313, 301), where u_0 and v_0 are the pixel coordinates of the principal point and f_u and f_v are the focal lengths in each principal direction. The initial camera position is (180.5, -10.49, 464.8) mm and orientation in Euler angles (α, β, γ) = (2.92, 0.02, 1.78) rad. The real camera is fixed at position (3.6, 39.7, 303.3) mm and orientation in Euler angles (α, β, γ) = (2.9, 0.8, 1.5) rad. Figure 1 shows the evolution of the mean error, $\mathbf{e}$, during the virtual visual servoing task. In this experiment we can see that the error decreases exponentially. In Figure 2, the virtual camera trajectory obtained during the virtual visual servoing task is represented. Once the task ends, the extrinsic parameters obtained are: position (9.95, 42.35, 302.3) mm and orientation in Euler angles (α, β, γ) = (2.98, 0.85, 1.57) rad. This position is very close to the real one. Therefore, it is possible to affirm that the virtual visual servoing approach can be employed to determine the camera extrinsic parameters during the task.

3. Time-independent method to track trajectories

3.1. Notation and trajectory to be tracked

In this paper, the presence of a planner, which provides the robot with the 3D trajectory, γ(t), to be tracked (i.e. the desired 3D trajectory of the camera at the end-effector), is assumed. For this study we have employed planners developed in our previous works [9].

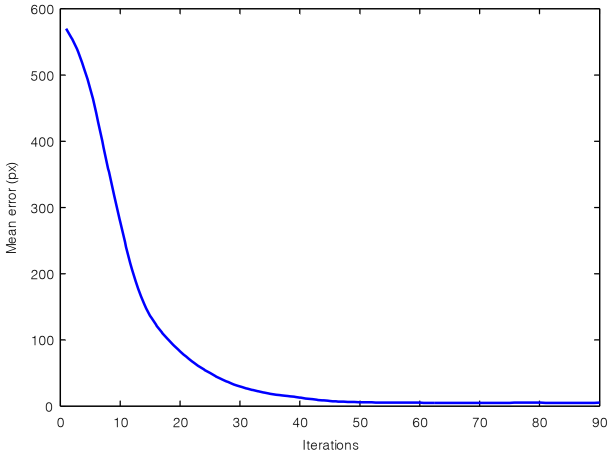

Figure 1. Mean error in pixel.

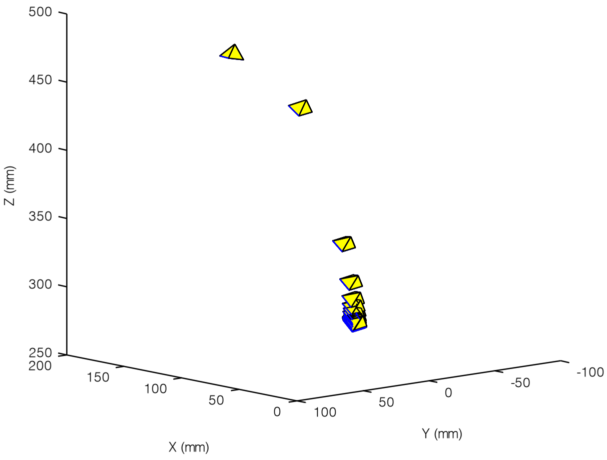

Figure 2. Camera trajectory.

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

$$\mathbf{A} = \begin{bmatrix} f \cdot p_u & -f \cdot p_u \cdot \cot(\theta) & u_0 \\ 0 & f \cdot p_v / \sin(\theta) & v_0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (7)$$

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

From τ the discrete trajectory of the features in the image is obtained $\boldsymbol{T} = \left\{ {}^{k}\mathbf{s} / k \in 1..N \right\}$, with ${}^{k}\mathbf{s}$ being the set of M points or features observed by the camera at instant k, ${}^{k}\mathbf{s} = \left\{ {}^{k}\boldsymbol{f}_i / i \in 1..M \right\}$. Figure 3 shows an example of 3D trajectory and Figure 4 shows the image trajectory in order to illustrate the previous mentioned notation.

3.2. Previous time-independent methods to track trajectories based on visual servoing

In our previous works we have developed a time-independent tracking system based on image-based visual servoing [10]. This approach, called movement flow-based visual servoing, employs the concept of movement flow. The movement flow is a vector field that converges towards the image trajectory. 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. This way, it allows the tracking of the trajectory.

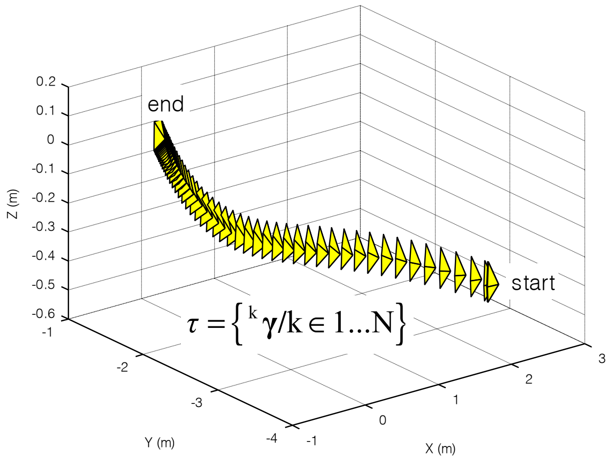

Figure 3. Notation employed: 3D trajectory to be tracked, $\tau = \left\{ {}^{k}\gamma / k \in 1...N \right\}$.

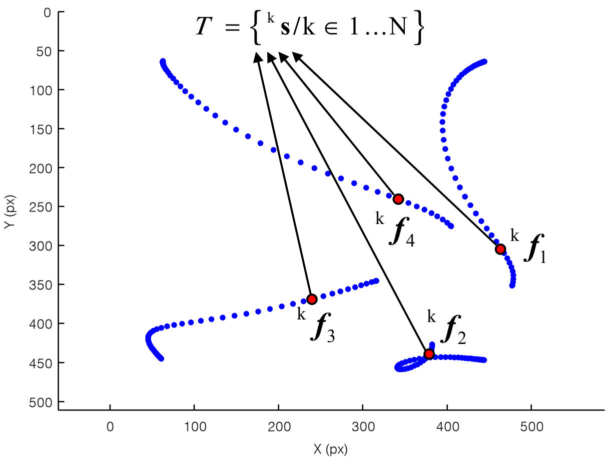

Figure 4. Notation employed: Image trajectory to be tracked, $\boldsymbol{T} = \left\{ {}^{k}\mathbf{s} / k \in 1..N \right\}$.

Hence, considering image-based control, the velocity applied to the robot with respect to the camera coordinate frame will be:

$$\boldsymbol{v}_M^C = -\lambda_M \cdot \hat{\mathbf{J}}_f^{+} \cdot \boldsymbol{e}_f \qquad (8)$$

where λ_M is the gain of the proportional controller; $\hat{\mathbf{J}}_f^{+}$ is the estimated pseudo-inverse of the interaction matrix [3], $\boldsymbol{e}_f = \mathbf{s} - \mathbf{s}_d$, $\mathbf{s}=[\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_M]^T$ are the features extracted from the image. Furthermore, the desired features will be $\mathbf{s}_d=[\boldsymbol{f}_1+m_1\Phi_1(\boldsymbol{f}_1), \boldsymbol{f}_2+m_2\Phi_2(\boldsymbol{f}_2), \ldots, \boldsymbol{f}_M+m_M\Phi_M(\boldsymbol{f}_M)]^T$, where $\boldsymbol{\Phi}_i$ is the movement flow for the feature i and $\boldsymbol{m}=\{m_1, m_2, \ldots, m_M\}$ determines the progression speed. This approach has several problems which must be solved in order to be applied in a real workspace. The movement flow-based visual servoing system has a correct behaviour in the 3D space. However, when the velocity is increased this fact is not guaranteed. Moreover, it is not possible to specify the desired velocity to do the tracking. Another problem is that the system presents an oscillating behaviour when difficult trajectories are being tracked. In order to solve these problems the method described in the next section is proposed.

3.3. Proposed method

In Figure 5, the different components of the proposed visual servoing system are represented. With ${}^{1}\mathbf{s}$ the visual features observed at the initial camera position are represented. From this initial set of image features it is necessary to find an image configuration which provides the robot with the desired velocity, $|\mathbf{v}_d|$. To do so, the system iterates over the set $\boldsymbol{T}$. For each image configuration ${}^{k}\mathbf{s}$ the corresponding camera velocity is determined considering an image-based visual servoing system (at this first stage $\mathbf{s} = {}^{1}\mathbf{s}$):

$$ {}^{k}\mathbf{v} = -\lambda_1 \hat{\mathbf{L}}_s^{+}\left(\mathbf{s} - {}^{k}\mathbf{s}\right) \tag{9} $$

This process continues until $|{}^{k}\mathbf{v}|$ is greater than the desired velocity, $|\mathbf{v}_d|$. At this moment, the set of features ${}^{k}\mathbf{s} = {}^{i}\mathbf{s}$ will be the desired features to be used by an image-based visual servoing system (see Equation (2)). However, the visual features which provide the desired velocity are between ${}^{k}\mathbf{s}$ and ${}^{k-1}\mathbf{s}$. To obtain the correct image features the method described in next section is proposed.

Therefore, once the control law represented in Equation (2) is executed the system searches again a new image configuration which provides the desired velocity. This process continues until the complete trajectory is tracked.

3.4. Trajectory interpolation to obtain the desired image features

In order to obtain the desired velocity during the tracking, $|\mathbf{v}_d|$, it is necessary to find a 3D camera location, ${}^{c}\mathbf{M}_{oj}$, from which a set of features ${}^{j}\mathbf{s}$ are extracted, so that applying the control law (2) the velocity $|\mathbf{v}_d|$ is obtained. As it is described in the past section, these visual features will be extracted between the features ${}^{k}\mathbf{s}$ and ${}^{k-1}\mathbf{s}$. The problem addressed in this section is to determine the 3D location ${}^{c}\mathbf{M}_{oj}$.

The camera extrinsic parameters ${}^{c}\mathbf{M}_{ok}$ and ${}^{c}\mathbf{M}_{ok-1}$ are obtained from the set of features ${}^{k}\mathbf{s}$ and ${}^{k-1}\mathbf{s}$ respectively by using virtual visual servoing (see Section 2). From ${}^{c}\mathbf{M}_{ok}$ and ${}^{c}\mathbf{M}_{ok-1}$ the translations ${}^{c}\mathbf{t}_{ok}$ and ${}^{c}\mathbf{t}_{ok-1}$ can be easily extracted. A linear interpolation between both translations can be obtained using the following equation:

$$ {}^{c}\mathbf{t}_{o\alpha} = \alpha\cdot\left({}^{c}\mathbf{t}_{ok} - {}^{c}\mathbf{t}_{ok-1}\right) \tag{10} $$

with $0 \le \alpha \le 1$. Furthermore, from ${}^{c}\mathbf{M}_{ok}$ and ${}^{c}\mathbf{M}_{ok-1}$ the rotations ${}^{c}\mathbf{R}_{ok}$ and ${}^{c}\mathbf{R}_{ok-1}$ can also be extracted. In order to develop the linear interpolation for the orientation it is first necessary to represent the previous rotations by the quaternions ${}^{c}\mathbf{Q}_{ok}$ and ${}^{c}\mathbf{Q}_{ok-1}$. In order to interpolate the orientation between the two quaternions, spherical linear interpolation is used as described in [12], obtaining ${}^{c}\mathbf{Q}_{o\alpha}$.

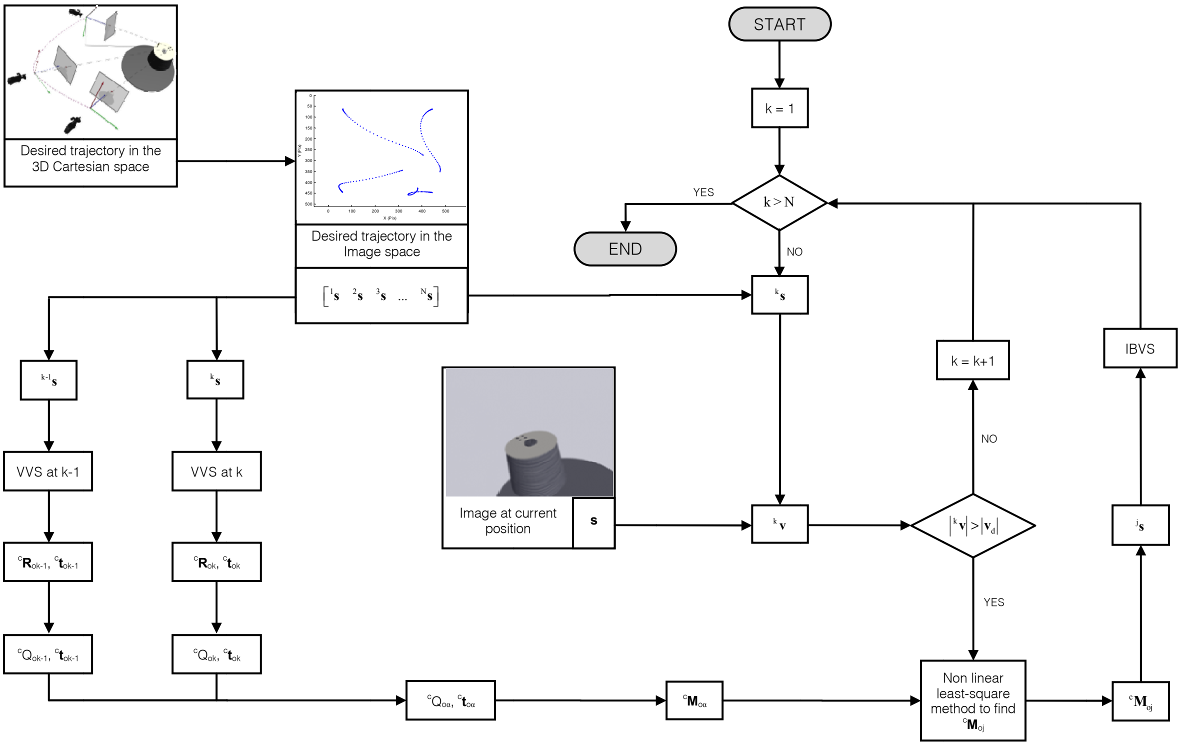

Figure 5. Scheme of the algorithm to track trajectories in the image space using visual servoing.

Once the linear interpolation ${}^{c}\mathbf{M}_{o}(\alpha)$ = $[{}^{c}\mathbf{R}_{o\alpha}\ {}^{c}\mathbf{t}_{o\alpha}]$ is generated, the location ${}^{c}\mathbf{M}_{oj}$ is obtained by non linear least squares optimization. To do so it is necessary to minimize the error $\mathbf{e}_{o} = \left|\mu_{c}(\alpha)\right| - \left|v_{d}\right|$ where:

$$\mu_{c}(\alpha) = -\lambda_{1}\hat{\mathbf{L}}_{s}^{+}\left(\mathbf{s} - \mathbf{s}^{*}(\alpha)\right) \qquad (11)$$

and

$$\mathbf{s}^{*}(\alpha) = pr_{\xi}\left({}^{c}\mathbf{M}_{o}(\alpha)\,{}^{o}\mathbf{P}\right) \qquad (12)$$

where ${}^{o}\mathbf{P}$ are the set of object positions considered and pr_{ξ} denotes the perspective projection model according to the intrinsic parameters, ξ.

Once the value α which minimizes $\mathbf{e}_{o}$ is obtained, the set of features ${}^{j}\mathbf{s}$ will be equal to $\mathbf{s}^{*}(\alpha)$.

4. Results

In this section, different results from the application of the proposed visual servoing system to track trajectories in the image space are described. In order to clearly describe the advantages of the algorithm presented in section 3, the tracking is firstly developed using a classical image-based visual servoing system. Subsequently, in order to show the correct behaviour of the proposed system, the same path is tracked with the method described in section 3. Finally, the movement flow-based visual servoing system is tested to show that the proposed method is able to track in a more precise way the previously planned path even in conditions of high velocities.

In the next section, the experimental setup employed is described.

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. Trajectory tracked by a classical Image-Based Visual Servoing.

The 3D trajectory τ to track is the one shown in Figure 3. With this 3D trajectory, the desired trajectory *T* in the image is the obtained in Figure 4.

Firstly, a classical image-based visual servoing system in which the desired features (see Equation (2)) are the ones observed at ${}^{N}\gamma$, i.e., ${}^{N}\mathbf{s}$ is considered. The behaviour of the image-based visual servoing is not the desired, not only in the 3D space (see Figure 6), but also in the image (see Figure 7). The trajectory of a given feature tends to follows a straight line. Therefore, this method is not adequate to track trajectories and can only be used to position the camera with respect to the observed object.

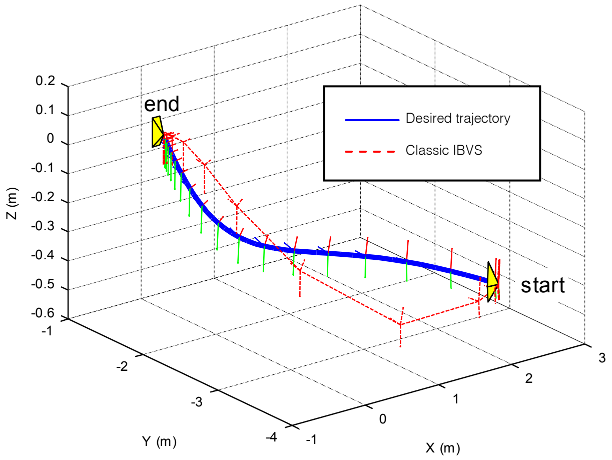

Figure 6. 3D trajectory of the camera using classic Image-Based Visual Servoing

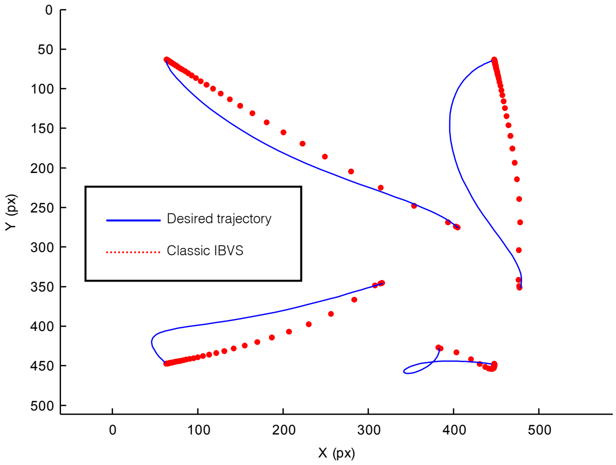

Figure 7. Image trajectory of the camera using classic Image-Based Visual Servoing

4.3. Trajectory tracked by the proposed method.

In this section, the proposed method is applied to track the trajectory shown in Figure 4. The method is able not only to track the desired trajectory, but also maintaining a desired velocity $\left|\mathbf{v}_{d}\right|$.The results obtained, considering the desired velocity $\left|\mathbf{v}_{d}\right| = 10\ \text{mm/s}$, are presented in Figure 8 and Figure 9. These show respectively, the 3D camera trajectory and the evolution of the features in the image. In these figures it is possible to observe that a correct tracking is obtained not only in the image, but also in the 3D space.

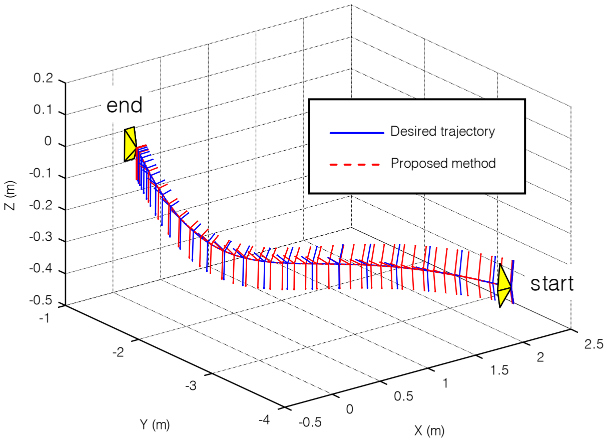

Figure 8. 3D trajectory of the camera using the proposed method

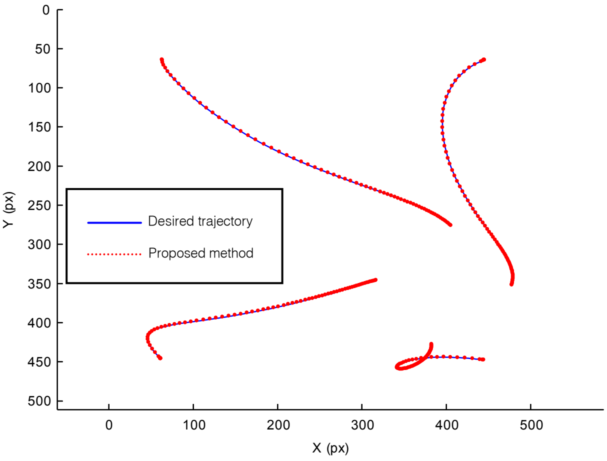

Figure 9. Image trajectory of the camera using the proposed method

Figure 10 shows the camera velocity during the tracking. As expected, the average magnitude of the translational and rotational velocities are approximately equal to 10 mm/s (see Figure 11).

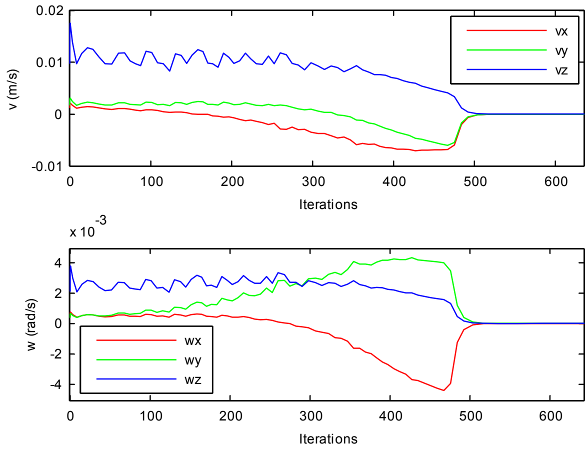

Figure 10. Linear and angular velocities of the camera along the tracking using the proposed method

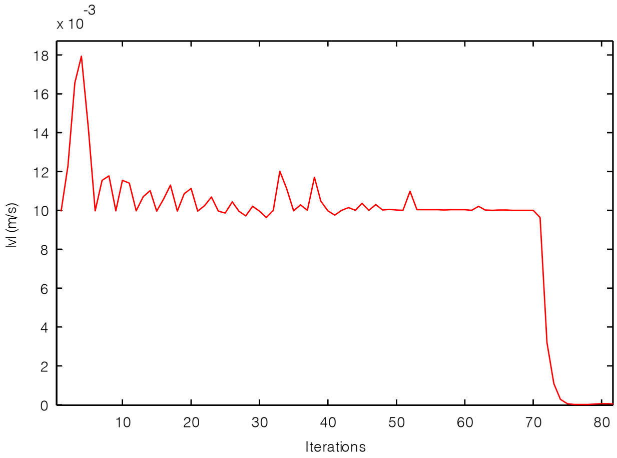

Figure 11. Magnitude of the linear velocity of the camera along the tracking using the proposed method

4.4. Movement flow-based visual servoing versus the proposed method.

Movement flow-based visual servoing [10] is a time-independent method to track trajectories in the image space using a visual servoing system. However, this method presents problems when high velocities are required for the tracking. In addition, it is not possible to specify the velocity for the tracking. If the tracking is developed with a high velocity, the movement flow-based visual servoing is not able to obtain a correct behaviour, neither in 3D nor in image space. Furthermore, using this system an oscillating behaviour is obtained.

To show the correctness of the proposed method in situations in which the movement flow-based visual servoing system does not work correctly, the image trajectory shown in Figure 12 has been tracked using both methods. Due to the fact that in movement flow-based visual servoing the velocity cannot be directly controlled, this method is firstly applied to track the desired trajectory. The parameters of this method are set in such a way that tracking velocity is high. Once the average velocity magnitude is obtained by tracking the trajectory with movement flow-based visual servoing, this velocity is specified to the proposed method in order to compare both results in similar conditions.

In this experiment, velocity using movement flow-based visual servoing is very near to $|\mathbf{v}| = 100$ mm/s. In order to compare both methods, this desired velocity is used in the proposed method to obtain good results.

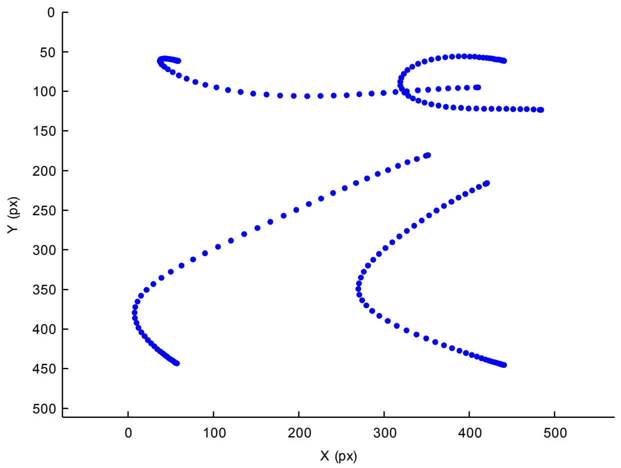

Figure 12. Desired image trajectory

In Figure 13, 3D trajectories followed by the camera using the two methods are represented. The desired trajectory to be tracked is also represented. Hence, the correct behaviour of the proposed method in 3D can be observed, as well as the poor performance of the movement flow-based visual servoing in the same conditions of high velocity. Using the proposed method, the deviation between desired trajectory and the obtained with it is small despite the high tracking velocity.

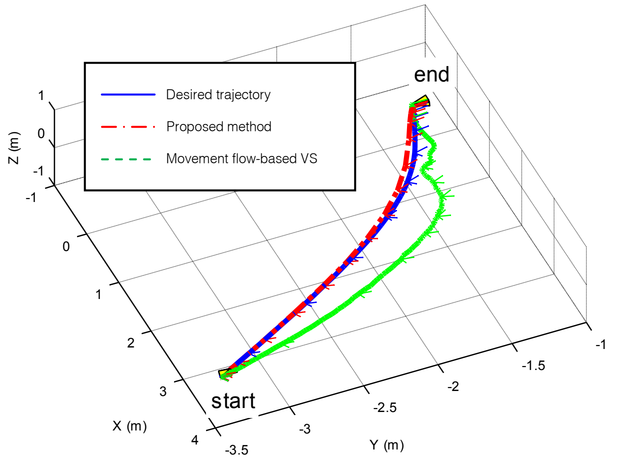

Figure 13. 3D position of the camera along the track using the proposed method and movement flow-based visual servoing

Figure 14 shows the trajectories followed by the four features in the image space. Again, in this figure the desired trajectory, the trajectory obtained using movement flow-based visual servoing and the one obtained using the proposed method are represented. In this case, it seems that the behaviour of the trajectory tracked by the proposed method is worse than the movement flow-based visual servoing one. Nevertheless, as Figure 13 demonstrates, the behaviour of the 3D trajectory is more accurate in the method proposed. This occurs because the proposed method guarantees that at each moment the camera presents a valid 3D configuration. This aspect is not guaranteed in movement flow-based visual servoing.

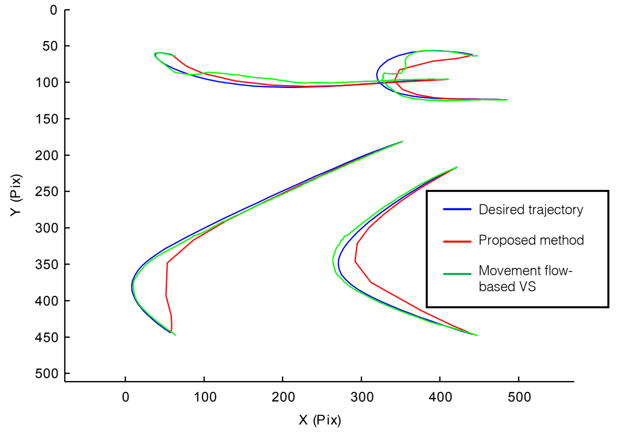

Figure 14. Image position of the features along the track using the proposed method and movement flow-based visual servoing

In addition, the method proposed here, not only has been proved to be better in 3D space, but also has the advantage of controlling the velocity to a constant desired value during the tracking. This property is shown in Figure 15 and Figure 16. In Figure 15 the linear and angular velocity during the tracking of the planned path using movement flow-based visual servoing has been represented. It can be seen that this velocity is very near to $|\mathbf{v}| = 100$ mm/s . Furthermore, this velocity can really be neither precisely nor directly controlled.

In Figure 16, linear and angular velocities during the tracking using the proposed method are shown. This method allows one to directly specify the desired tracking velocity. The tracking has been carried out using a desired velocity of $|\mathbf{v}| = 100$ mm/s . Therefore, the image trajectory has been tracked with approximately the same velocity as the tracked using the movement flow-based visual servoing.

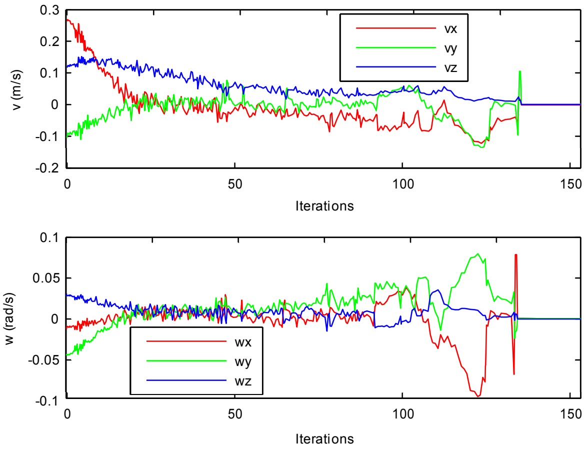

Figure 15. Linear and angular velocities of the camera along the tracking using movement flow-based visual servoing

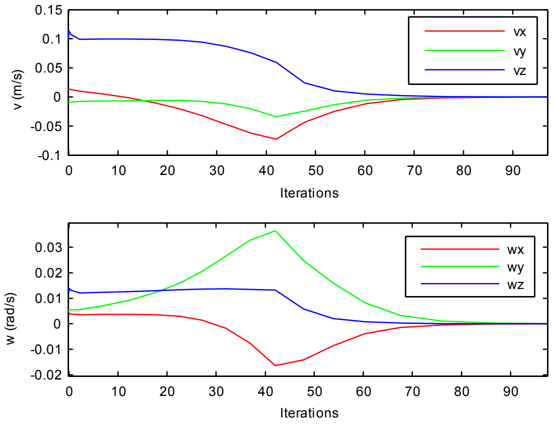

Figure 16. Linear and angular velocities of the camera along the tracking using the proposed method

Figure 16 shows that the velocity magnitude has been correctly controlled. Furthermore, as it can be seen in Figure 16, the tracking is performed with less iterations while the proposed method is used. Moreover, the trajectory is tracked with less oscillating behaviour than the obtained with the movement flow-based visual servoing (see Figure 15). Therefore, the proposed method to track trajectories based on visual servoing improves the behaviour of previous time-independent methods.

5. Conclusions

In this paper, a new method to track trajectories in the image space has been described. It allows the accurate tracking of a previously generated trajectory of the features in the image space. The method has a time-independent behaviour. In addition, this proposed method allows us to specify the camera velocity along the track.

Experimental results show the validity of the proposed algorithm to track trajectories in the image space. In these results, it can be observed that previous time-independent methods like movement flow-based visual servoing fail when the tracking is carried out with a high velocity. The proposed method allows not only controlling the velocity of the tracking, but also obtaining less oscillating behaviour and needs a smaller number of iterations to end the task. Furthermore, experimental results show that the 3D position of the camera during the tracking results in less error using the proposed method. The approach to the desired trajectory is more precise.

Acknowledgements

This work was funded by the Spanish MEC project "Diseño, implementación y experimentación de escenarios de manipulación inteligentes para aplicaciones de ensamblado y desensamblado automático".

References

[1] F. Chaumette and S. Hutchinson. "Visual Servo Control, Part I: Basic Approaches", *IEEE Robotics and Automation Magazine*, 13(4):82-90, 2006.

[2] F. Chaumette, "Potential problems of convergence in visual servoing", *Int. Symposium on Mathematical Theory of Networks and Systems,* Padoue, Italy, 1998.

[3] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control", *IEEE Trans. Robotics and Automation*, vol. 12, no. 5, pp. 651-670, Oct, 1996.

[4] K. Kinoshita, K. Deguschi, "Simultaneous determination of camera pose and intrinsic parameters by visual servoing", *Proc. 12th International Conference on Pattern Recognition*, Jerusalem, Israel, vol. A, pp. 285-289, 1994.

[5] E. Marchand and F. Chaumette, "Virtual visual servoing: a framework for real-time augmented reality", *Proc. Eurographics 2002*, Saarebrücken, Germany, vol. 21, no. 3, pp. 289-298, 2002.

[6] E. Marchand and F. Chaumette, "A new formulation for non-linear camera calibration using VVS", *Publication Interne 1366, IRISA,* Rennes, France, 2001.

[7] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control", *IEEE Transactions on Robotics and Automation*, 18 (4), pp. 534-549, 2002.

[8] Y. Mezouar, A. Remazeilles, P. Gros and F. Chaumette, "Images interpolation for image-based control under large displacement", *IEEE International Conference on Robotics and Automation*, 4, pp. 3787-3794, 2002.

[9] J. Pomares, P. Gil, G. J. García and F. Torres. "Visual–Force Control and Structured Light Fusion to Improve Recognition of Discontinuities in Surfaces", *11th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA),* Prague, 2006.

[10] J. Pomares and F. Torres, "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, Jan, 2005.

[11] F. Schramm and G. Morel. "Ensuring visibility in calibration-free path planning for image-based visual servoing", *IEEE Transactions on Robotics*, 22 (4), pp. 848-854, 2006.

[12] K. Shoemake, "Animating Rotation with Quaternion Curves", *Computer Graphics* V.19 N.3, 1985.

[13] R. Y. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses", *IEEE J. Robotics Automat.*, vol. RA-3, no. 4, pp. 323-344, 1987.

[14] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations", *International Conference on Computer Vision*, Corfu, Greece, 1999, pp. 666-673.