

# Automatic robotic tasks in unstructured environments using an image path tracker

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

## Abstract

*This paper describes a new method to allow a robot to perform automatic tasks in an unstructured environment. To show that this method works properly, a task to replace a blown light bulb in a streetlamp is described. In order to perform this task correctly, the robot is positioned following secure previously defined paths. The robot, using an eye-in-hand configuration on a visual servoing scheme and a force sensor, is able to interact with its environment due to the fact that the path tracking is performed with time-independent behaviour. The desired path is given in the image. However, the proposed method obtains a correct tracking not only in the image, but also in the 3D space. This method solves the problems of the previously proposed time-independent tracking systems based on visual servoing, such as the possibility of specifying the desired tracking velocity, less oscillating behaviour and a correct tracking in the 3D space when high velocities are used. The experiments shown demonstrate the necessity of time-independent behaviour in tracking and the correct performance of the system.*

*Keywords: visual servoing, path tracking, temporal restrictions, robot control, automation.*

## 1. Introduction

This paper presents a strategy to resolve the problem of automatically replace a faulty bulb in a streetlamp. The task is carried out by a robotic arm in a real and unstructured environment like the final location of the streetlight in a public zone.

In a robotic task, the robot must be frequently positioned at a fixed location with respect to the objects in the scene. However, the position of these objects is not always known precisely. So, it is not possible to previously assure the location of the end-effector of the robot to accomplish the task correctly. Visual servoing is a technique that permits to position a robot with respect to an object using visual information (Hutchinson, Hager, & Corke, 1996). Basically, it consists of extracting visual data from an image acquired from a camera and comparing it with the visual data obtained at the desired position of the robot. By minimizing the error between the two images it is possible to move the robot to the desired position. Image-based visual servoing uses only the visual data obtained in an image to control the robot movement. These systems do not require computing the 3D position of the object. Therefore, image-based visual servoing systems are robust with respect to calibration errors in the camera and in the robot. The behaviour of these systems has been proved to be robust in local conditions (i.e., in conditions in which the initial position of the robot is very near to its final location) (Chaumette & Hutchinson, 2006). However, in large displacements, the errors in the computation of the intrinsic parameters of the camera have influence on the correct behaviour of the system (Chaumette, 1998). When the camera calibration parameters are not known correctly, a non-linear evolution of the features in the image space is obtained and some image features can disappear. By employing this basic image-based visual servoing scheme, it is possible to specify a target position. However, the 3D trajectory described by the robot to achieve the target position cannot be indicated. This behaviour is not adequate to the considered task of automatically changing a light bulb. The robot must be able to avoid other fixed objects located inside its workspace following a desired 3D path precisely. Moreover, the robot is located in a real and

unstructured workspace where the robot can be obstructed by any unexpected object located in its trajectory.

When a robot interacts with its environment, most of the methods to track image trajectories proposed up to now fail (Mezouar & Chaumette, 2002; Schramm & Morel, 2006). 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.

To solve the time-variable reference problem previously addressed, in Pomares and Torres (2005), a time-independent method to track trajectories called movement flow-based visual servoing is presented. However, this method presents 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 (Chaumette, 1998). 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.

The method proposed here has no temporal references. The time-independent behaviour of the system is crucial to permit interactions between the robot and its workspace. The new method described in this paper guarantees the correct tracking in the 3D space. To do so, 3D information obtained from the on-line calibration of the camera is used. Furthermore, the proposed system permits to specify a constant desired velocity during the tracking. The proposed method modifies classical image-based visual servoing in order to obtain a control strategy able to track image trajectories. Deriving the method from the classical image-based visual servoing permits to take advantage of the results and the background obtained in the last years in this field.

Over the last few decades, camera calibration methods have been widely investigated (see e.g. Tsai (1987); Zhang (1999)). Within these methods, virtual visual servoing systems have been a subject of recent researches (Kinoshita & Deguschi, 1994; Marchand & Chaumette, 2002). 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. In Chesi and Hashimoto (2004), a simple technique for camera displacement estimation in visual servoing tasks is presented. This estimation method does not require previous knowledge of the Euclidean structure of the object. However, this method requires to previously calibrating the camera. Virtual visual servoing not only is useful to compute the location of the camera, but also the intrinsic parameters of it.

This paper is organized as follows: the employed notation and the desired trajectory to be tracked are described in Section 2; Section 3 recalls the basic virtual visual servoing approach; Section 4 describes the proposed visual servoing scheme to track trajectories; in Section 5 the proposed tracking algorithm is compared with the previous visual servoing tracking systems; Section 6 presents the experimental results for the change of a blown light bulb employing the proposed tracking method; and the final section presents the main conclusions.

## 2. Notation and trajectory to be tracked

In this paper, the presence of a planner, which provides the robot with the 3D trajectory,  $\gamma(t)$ , to be tracked (i.e. the desired 3D trajectory of the camera located at the end-effector), is assumed. For this study planners developed in previous works have been employed (Pomares, Gil, Garcia, & Torres, 2006).

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  $\tau = \{^k \gamma / k \in 1 \dots N\}$ , is obtained.

${}^k \mathbf{P}_i / i \in 1 \dots M$  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  ${}^k\mathbf{p}_i = pr_{\xi}({}^c\mathbf{M}_0, {}^k\mathbf{P}_i)$ , where  $pr_{\xi}$  denotes the perspective projection model according to the intrinsic parameters,  $\xi$ , and  ${}^c\mathbf{M}_0$  represents the extrinsic camera parameters, i.e., the pose of the object frame with respect to the camera frame.

From  $\tau$  the discrete trajectory of the features in the image is obtained,  $\mathbf{T} = \{{}^k\mathbf{s} / k \in 1..N\}$ , with  ${}^k\mathbf{s}$  being the set of  $M$  points or features observed by the camera at instant  $k$ ,  ${}^k\mathbf{s} = \{{}^k\mathbf{p}_i / i \in 1..M\}$ . Figure 1 shows an example of 3D trajectory and Figure 2 shows the image trajectory in order to illustrate the previous mentioned notation.

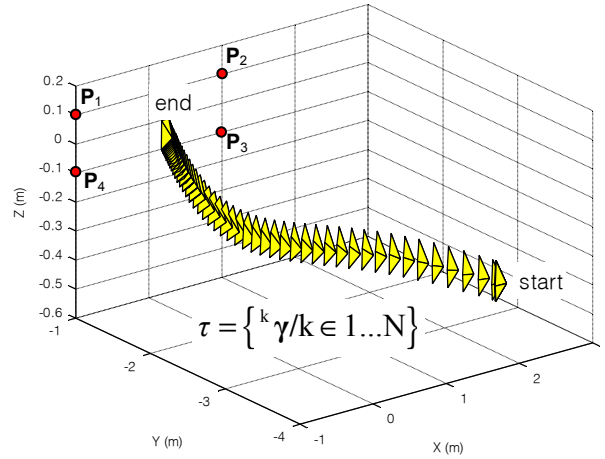


Figure 1. Notation employed: 3D trajectory to be tracked,  $\tau = \{{}^k\gamma / k \in 1..N\}$ .

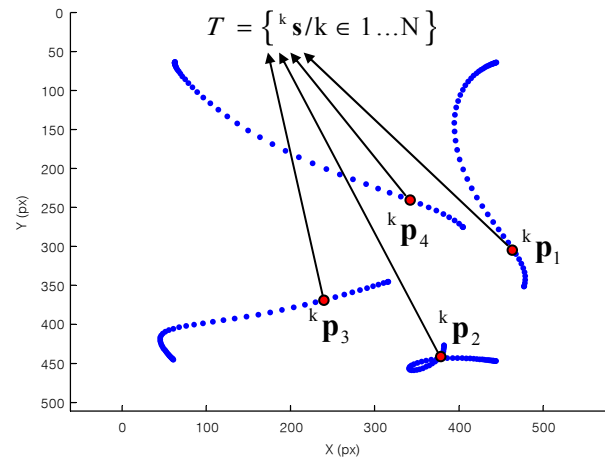


Figure 2. Notation employed: Image trajectory to be tracked,  $\mathbf{T} = \{{}^k\mathbf{s} / k \in 1..N\}$ .

### 3. 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:

$$\mathbf{e}_t = \mathbf{s} - \mathbf{s}^* \quad (1)$$

where  $\mathbf{s}$  is a  $M \times 1$  vector containing  $M$  visual features corresponding to the current state, while  $\mathbf{s}^*$  denotes the visual features values in the desired state.

With  $\mathbf{L}_s$  is represented the interaction matrix which relates the variations in the image with the variation in the velocity of the camera (Chaumette & Hutchinson, 2006):

$$\dot{\mathbf{s}} = \mathbf{L}_s \cdot \dot{\mathbf{r}} \quad (2)$$

where  $\dot{\mathbf{r}}$  indicates the velocity of the camera.

By imposing an exponential decrease of  $\mathbf{e}_t$  ( $\dot{\mathbf{e}}_t = -\lambda \mathbf{e}_t$ ) it is possible to obtain the following control action for a classical image-based visual servoing:

$$\mathbf{v}_c = -\lambda \hat{\mathbf{L}}_s^+ (\mathbf{s} - \mathbf{s}^*) \quad (3)$$

where  $\hat{\mathbf{L}}_s^+$  is the pseudoinverse of an approximation of the interaction matrix (Chaumette & Hutchinson, 2006), and  $\lambda$  is the proportional control gain.

The virtual visual servoing approach is based on the previous mentioned image-based visual servoing scheme. In virtual visual servoing, a virtual camera is positioned in the scene. From an image acquired by a real camera (the reference image), virtual visual servoing works moving the virtual camera iteratively. The goal is to obtain the location of the virtual camera from which the object is observed at the same position as in the reference image. In order to define the virtual visual servoing systems, the observed features in the reference image are denoted by  $\mathbf{p}_d$ , and  $\mathbf{p}$  are the

current positions of the image features projected using the camera intrinsic parameters,  $\xi$ , 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:

$$\mathbf{p} = pr_{\xi} ({}^c\mathbf{M}_o \circ \mathbf{P}) \quad (4)$$

where  ${}^o\mathbf{P}$  are the 3D coordinates (with respect to the object coordinate frame) of the points extracted by the camera and  $pr_{\xi}$  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):

$$\dot{\mathbf{e}} = \dot{\mathbf{p}} - \dot{\mathbf{p}}_d = \frac{\partial \mathbf{p}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} \quad (5)$$

where  $\mathbf{r}$  is the camera pose. Equation (5) can be rewritten as:

$$\dot{\mathbf{e}} = \mathbf{L}_p \mathbf{v} \quad (6)$$

where  $\mathbf{v}$  is the instantaneous virtual camera velocity and  $\mathbf{L}_p$  the interaction matrix related to  $\mathbf{p}$  (Marchand & Chaumette, 2001).

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

$$\mathbf{v} = -\lambda_1 \mathbf{L}_p^+ \mathbf{e} \quad (7)$$

Once  $\mathbf{e} = 0$  the extrinsic camera parameters are obtained. This basic scheme of the virtual visual servoing approach is enhanced in Pomares, Chaumette and Torres (2007), where virtual visual servoing is used to compute not only the extrinsic parameters of the real camera, but also the intrinsic ones. In this way, virtual visual servoing allows to obtain the calibration of the camera and its relative position with reference to an object in the image. This calibration can be achieved from

an image obtained by the real camera. In a visual servoing task, the use of virtual visual servoing before each iteration of the control loop allows to improve the system behaviour.

#### 4. Image path tracker based on visual servoing

In order to robustly place the robot with respect to a specific object, this section presents a new method to track a previously defined path in the image. First, a brief summary of previous systems in the literature to track trajectories in the image is presented, highlighting the disadvantages of these approaches. Then, the proposed system to track trajectories improving the behaviour of the methods proposed up to now is described. This new method requires computing an interpolated path between two references. To compute it, the method uses virtual visual servoing as described in the last subsection.

##### 4.1. Previous methods to track trajectories based on visual servoing

Up to now, the systems used to develop the tracking of a previously defined path using visual servoing are time-dependent (Malis, 2004; Mezouar & Chaumette, 2002), i.e. the references depend on the time:

$$\mathbf{v}_c = -\lambda \hat{\mathbf{L}}_s^+ (\mathbf{s} - \mathbf{s}^*(t)) \quad (8)$$

where the time varying vector  $\mathbf{s}^*(t)$  is the desired trajectory of the image features  $\mathbf{s}$ . This desired trajectory is obtained by spline interpolation of a set of features  $\mathbf{T}$ , considering the time between two consecutive frames constant.

So, if the robot interacts with an object placed in its workspace, the system continues sending references to it even though the robot cannot move. Once the obstruction ends, the references that have been sent up to the moment are lost, not allowing the correct tracking of the path.

There are several works in the literature dealing with obstacle avoidance when tracking a predefined path (Mansard & Chaumette, 2007; Soueres, Hamel, & Cadenat, 1998). This obstacle avoidance is not valid for the case discussed here. An obstacle avoidance subsystem needs



additional sensors to detect the obstacle, as in Soueres et al. (1998), where an ultrasonic sensor is employed. In Mansard and Chaumette (2007) the obstacle detector is not described due to the complexity of these systems, and the path is modified on-line but using the location of the obstacle obtained off-line. In this paper, additional sensors to avoid obstacles are not employed. Therefore, to track the path once the collision is detected, the non-temporal approach is employed.

Along this last line, in Schramm and Morel (2006), the visual servoing system proposed to track the trajectory in the image deals with the tracking task as a sequence of positioning tasks. It employs the classical image-based visual servoing control law (see Equation (3)) to position the robot between two consecutive positions into the path. Once the error is annulled (the robot velocity is 0), the system guides the robot to the next position in the path. This process is repeated until the robot has visited all the references in the path. Thus, this system is not able to maintain a constant velocity during the tracking (see Figure 3).

Using a time-dependent method to perform the tracking, it cannot be assured that the robot visits all the references (see experiment shown in Figure 15). If the robot is stopped by the obstacle and it cannot move until the object disappears, any of the obstacle avoidance methods with time-dependent behaviour can assure to resume the task. In a time-dependent system, once the obstruction ends the next temporal reference in  $s^*(t)$  is applied. So that depending on the duration of the obstruction, the system will continue the tracking of the trajectory at different points. This aspect is more clearly described in Section 5.3.

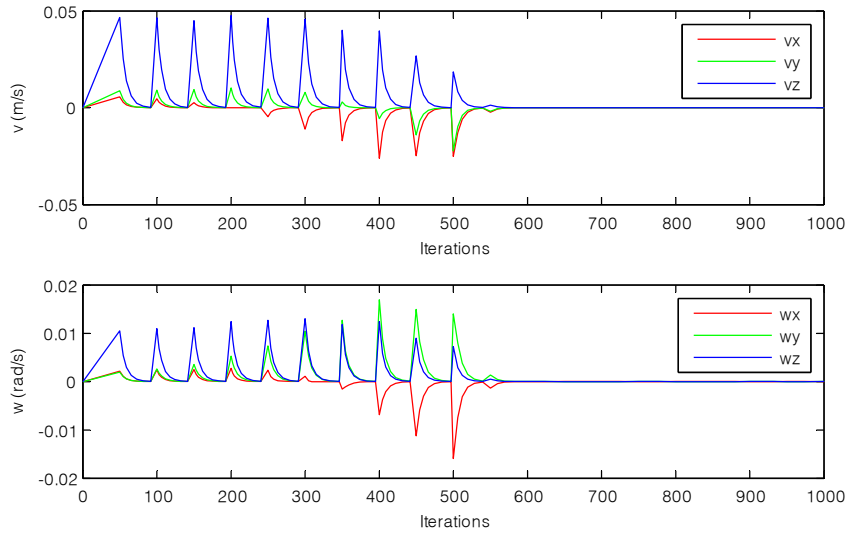


Figure 3. Variable velocity in the tracking using a standard image-based visual servoing.

In previous works a time-independent (the references do not depend on the time) tracking system based on image-based visual servoing has been presented (Pomares & Torres, 2005). This approach, called movement flow-based visual servoing, employs the concept of movement flow. 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. However, 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 behaviour is not guaranteed. Moreover, it is not possible to specify the desired velocity to do the tracking. Another problem is that the system has an oscillatory behaviour when difficult trajectories are being tracked. In order to solve these problems the method described in the next section is proposed.

#### 4.2. Time-independent method to track image paths using visual servoing based on virtual visual servoing

The method developed to track trajectories in the image must resolve the problems found in the methods proposed up to now. Firstly, it must present a time-independent behaviour to allow the

robot to interact with the environment. Secondly, it must improve the performance of movement flow based visual servoing, as it also has a time-independent behaviour. To do so, the proposed method must be able to control the desired tracking velocity.

With  ${}^1\mathbf{s}$  the set of 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  $\mathbf{T}$ . For each image configuration  ${}^k\mathbf{s}$  the corresponding camera velocity is determined considering an image-based visual servoing system (at this first stage  $\mathbf{s} = {}^1\mathbf{s}$ ):

$${}^k\mathbf{v} = -\lambda\hat{\mathbf{L}}_s^+(\mathbf{s} - {}^k\mathbf{s}) \quad (9)$$

This process continues until  $|{}^k\mathbf{v}|$  is greater than the desired velocity,  $|\mathbf{v}_d|$  (see Figure 4). At this moment, the set of features  ${}^k\mathbf{s}$  will be the desired features to be used by an image-based visual servoing system (see Equation (3)). Figure 5 summarizes this search process. 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.

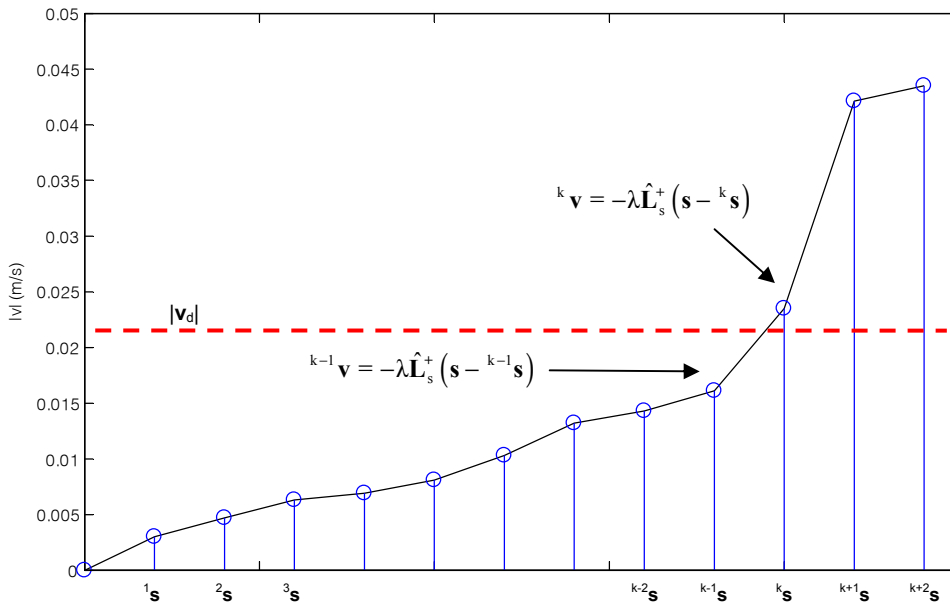


Figure 4. Set of image features greater than the desired velocity.

Therefore, once the control law represented in Equation (3) is executed the system searches again for a new image configuration which provides the desired velocity. This process continues until the complete trajectory is tracked.

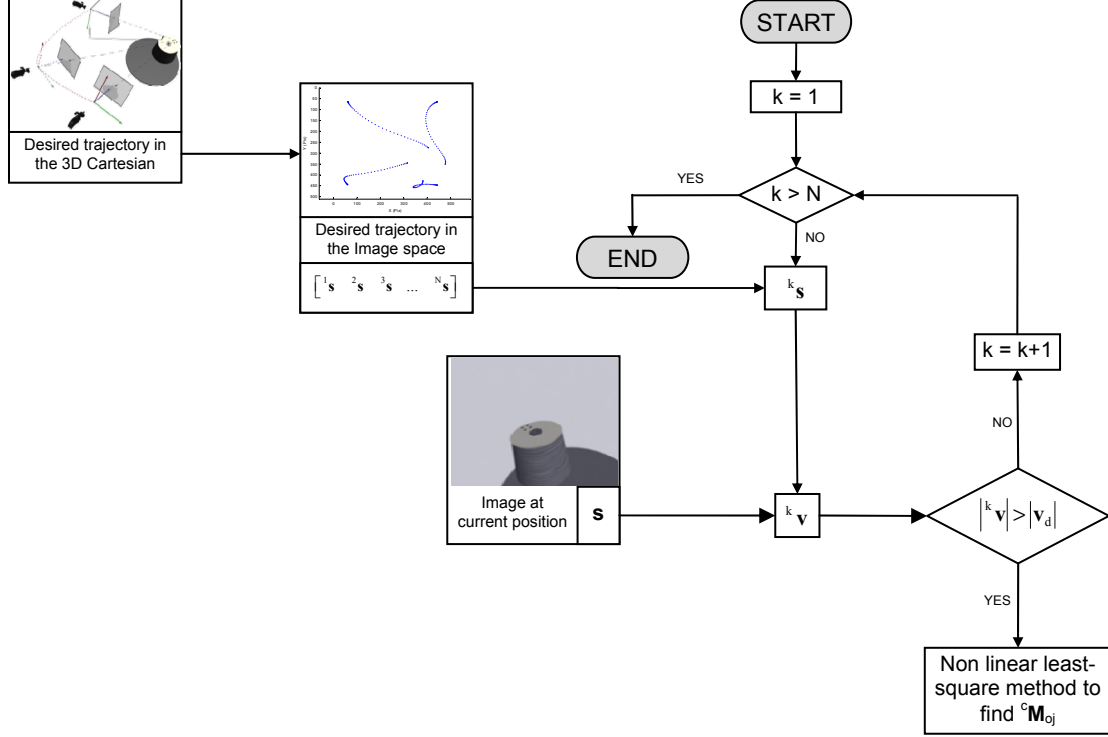


Figure 5. Scheme of the algorithm to search the set of features from which the velocity using an IBVS control law is greater than the desired one.

### 4.3. Trajectory interpolation to obtain the desired image features

In order to obtain the desired velocity during the tracking,  $|v_d|$ , it is necessary to find a 3D camera location,  ${}^cM_{oj}$ , from which a set of features  ${}^j s$  are extracted, so that applying the control law (3) the velocity  $|v_d|$  is obtained. As it is described in the past section, these visual features will be extracted between the set of features  ${}^k s$  and  ${}^{k-1} s$ . The problem addressed in this section is to determine the 3D location  ${}^cM_{oj}$ . Figure 6 schematizes the algorithm proposed to obtain this 3D location.

The camera extrinsic parameters  ${}^cM_{ok}$  and  ${}^cM_{ok-1}$  are obtained from the set of features  ${}^k s$  and  ${}^{k-1} s$  respectively by using virtual visual servoing (see Section 2). From  ${}^cM_{ok}$  and  ${}^cM_{ok-1}$  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:

$${}^c\mathbf{t}_{o\alpha} = {}^c\mathbf{t}_{ok-1} + \alpha \cdot ({}^c\mathbf{t}_{ok} - {}^c\mathbf{t}_{ok-1}) \quad (10)$$

with  $0 \leq \alpha \leq 1$ . 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 firstly it is 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 Shoemake (1985), obtaining  ${}^c\mathbf{Q}_{o\alpha}$ :

$${}^c\mathbf{Q}_{o\alpha} = \frac{{}^c\mathbf{Q}_{ok-1} \cdot \sin((1-\alpha)\theta) + {}^c\mathbf{Q}_{ok} \cdot \sin(\alpha\theta)}{\sin\theta} \quad (11)$$

where  $\theta$  is the angle between the orientation of the camera in k-1 and the orientation in k. After the computation of  ${}^c\mathbf{Q}_{o\alpha}$ , the rotation matrix  ${}^c\mathbf{R}_{o\alpha}$  is obtained by transforming now the quaternion.

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  $\mathbf{e}_o = |\mu_c(\alpha)| - |v_d|$

where:

$$\mu_c(\alpha) = -\lambda_1 \hat{\mathbf{L}}_s^+ (\mathbf{s} - \mathbf{s}^*(\alpha)) \quad (12)$$

and

$$\mathbf{s}^*(\alpha) = pr_{\xi} ({}^c\mathbf{M}_o(\alpha) \circ \mathbf{P}) \quad (13)$$

where  $\circ\mathbf{P}$  are the set of object points considered and  $pr_{\xi}$  denotes the perspective projection model according to the intrinsic parameters,  $\xi$ .

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

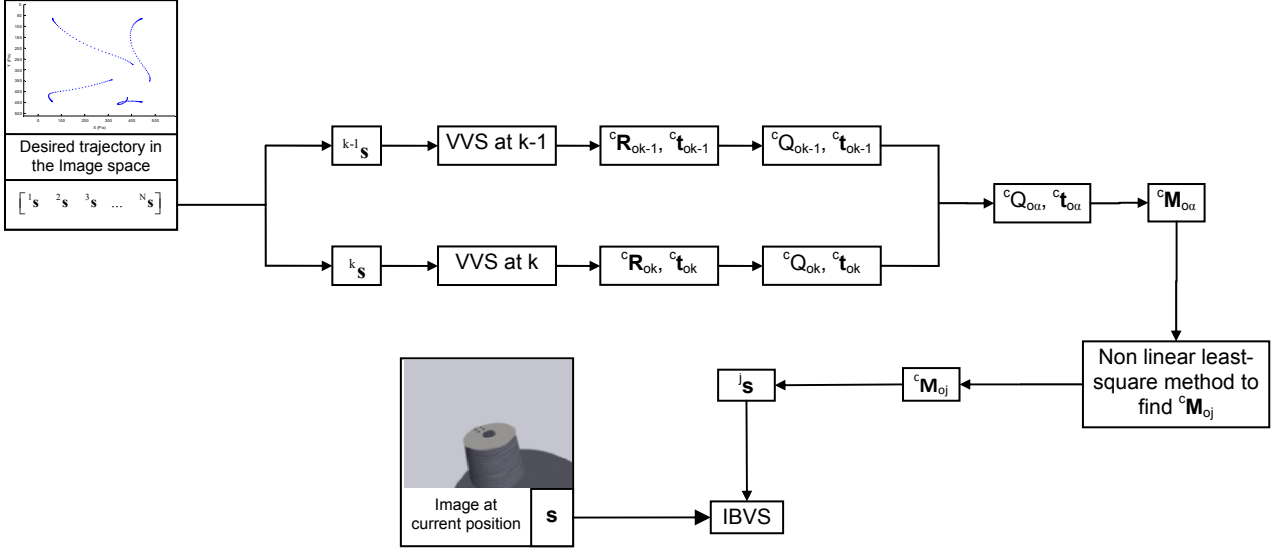


Figure 6. Scheme of the algorithm to interpolate the path obtaining the desired set of features that assures the desired velocity during the tracking.

## 5. Comparing the proposed time-independent method to track trajectories in the image using visual servoing with the previous systems

In order to clearly describe the advantages of the algorithm presented in section 4, the tracking is firstly developed using a classical image-based visual servoing system. Subsequently, to show the correct behaviour of the proposed system, the same path is tracked with the method described in section 4. Afterwards, the proposed method with a time-independent behaviour is compared with systems with temporal references. 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.

### 5.1. Trajectory tracked by a classical Image-Based Visual Servoing system.

The 3D trajectory  $\tau$  to track is the one shown in Figure 1. With this 3D trajectory, the desired trajectory  $T$  in the image is the one obtained in Figure 2.

Firstly, a classical image-based visual servoing system in which the desired features (see Equation (3)) are the ones observed at  $N\gamma$ , i.e.,  $\mathbf{s}^* = N\mathbf{s}$  is considered. The behaviour of the image-based visual servoing is not the desired, not only in the 3D space (see Figure 7), but also in the

image (see Figure 8). The trajectory of a given feature tends to follow 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.

Using the classical image-based visual servoing control law, only one reference can be loaded to the control loop (i.e., the system only is able to perform a positioning task). In this way, the evolution of the features in the image is not controlled during the visual servoing task. Figure 3 shows an experiment in which the tracking of the trajectory is obtained by the consecutive execution of the image-based visual servoing control law, modifying the desired image features. The problem here is that the velocity tracking is not constant. Basic image-based visual servoing scheme has been improved in the literature in order to permit not only to achieve positioning tasks, but also to perform a tracking of an image path. The different methods developed have been briefly described in Section 4.1. They can be divided into time-dependent and time-independent methods. In the next sections the proposed method (with a time-independent behaviour) is compared with previous methods to show the necessity of this new system which solves the problems detected in the methods employed up to now.

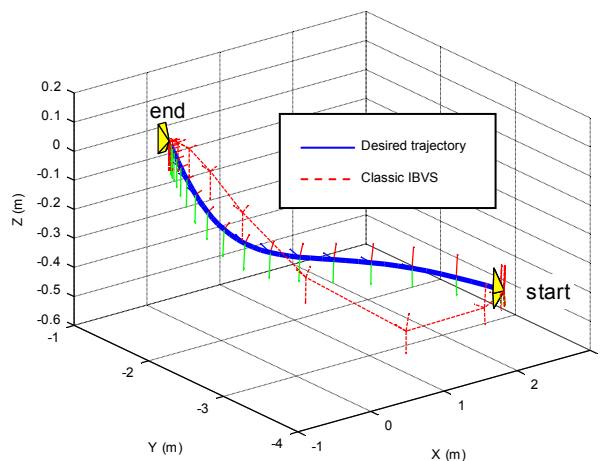


Figure 7. 3D trajectory of the camera using classic Image-Based Visual Servoing

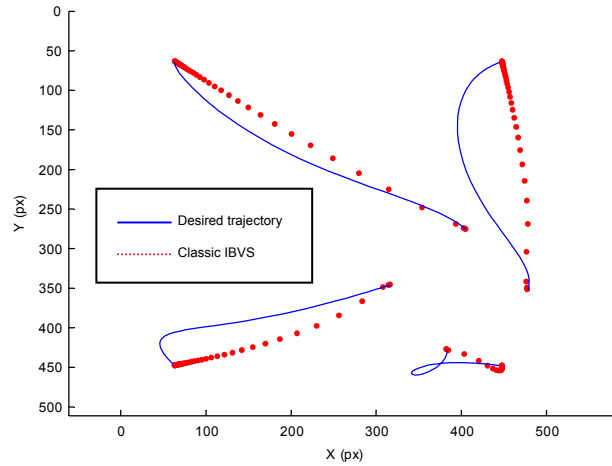


Figure 8. Image trajectory of the camera using classic Image-Based Visual Servoing

## 5.2. Trajectory tracked by the proposed method.

In this section, the proposed method is applied to track the trajectory shown in Figure 2. The method is able not only to track the desired trajectory, but also maintaining a desired velocity  $|\mathbf{v}_d|$ . The results obtained, considering the desired velocity  $|\mathbf{v}_d| = 10 \text{ mm/s}$ , are presented in Figure 9 and Figure 10. 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.

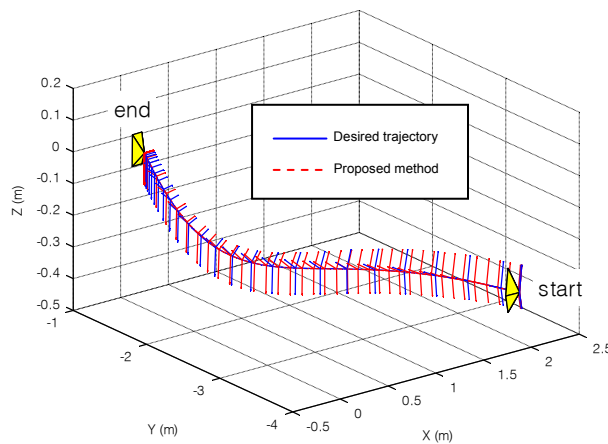


Figure 9. 3D trajectory of the camera using the proposed method



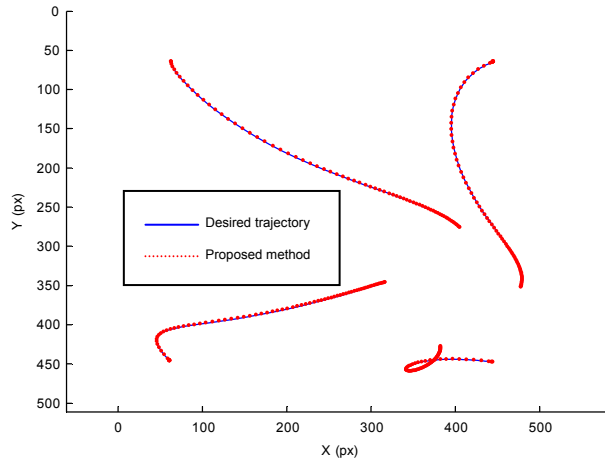


Figure 10. Image trajectory of the camera using the proposed method

Figure 11 shows the camera velocity during the tracking. As expected, the average magnitude of the translational and rotational velocities is approximately equal to 10 mm/s (see Figure 12). The time elapsed in a control loop iteration is 10 ms.

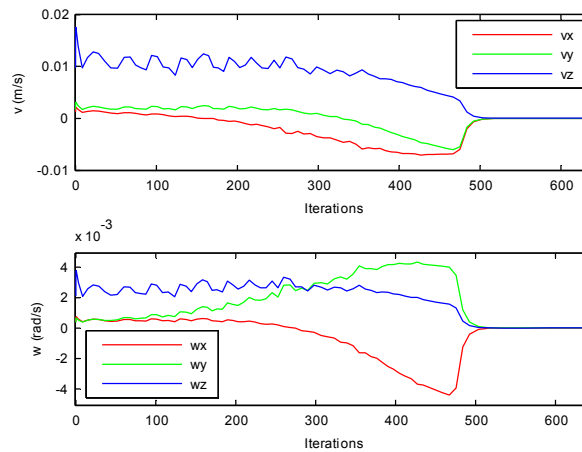


Figure 11. Linear and angular velocities of the camera along the tracking using the proposed method

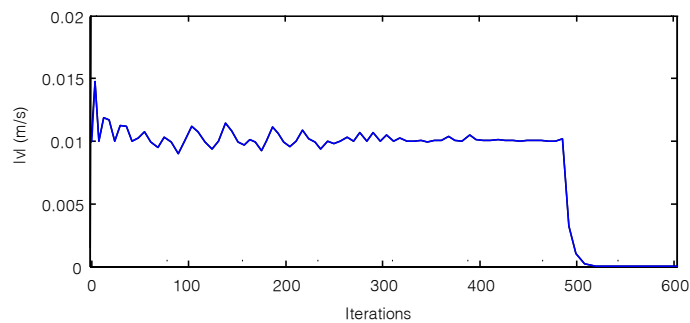


Figure 12. Magnitude of the linear velocity of the camera along the tracking using the proposed method

### 5.3. Time-dependent methods versus the proposed time-independent method.

In applications like the change of a faulty bulb in a streetlamp, the robot is working in a real environment in which it can interact with any object located in its workspace. This is the case for the change of a light bulb in a streetlamp located in its final position in the street. A force sensor is mounted at the end-effector of the robot to permit the interruption of the trajectory tracking when a high interaction force is detected. In these conditions the proposed system is employed to carry out the tracking of the desired image trajectory. If there is no interaction with the environment and the time-independent system to track trajectories in the image based on virtual visual servoing acts normally, the tracking takes 12 seconds. However, in order to expose the difference between the system behaviour and a time-dependent system, Figure 13 shows an example of a tracking in which within the trajectory the robot is obstructed for approximately 3 seconds. During the obstruction, the robot goes off the 3D desired path. In Figure 13 it can be observed that, once the obstruction ends, the system continues with the path tracking from the next reference point in the desired trajectory. Figure 14 shows the evolution of the features in the image using the proposed time-independent method.

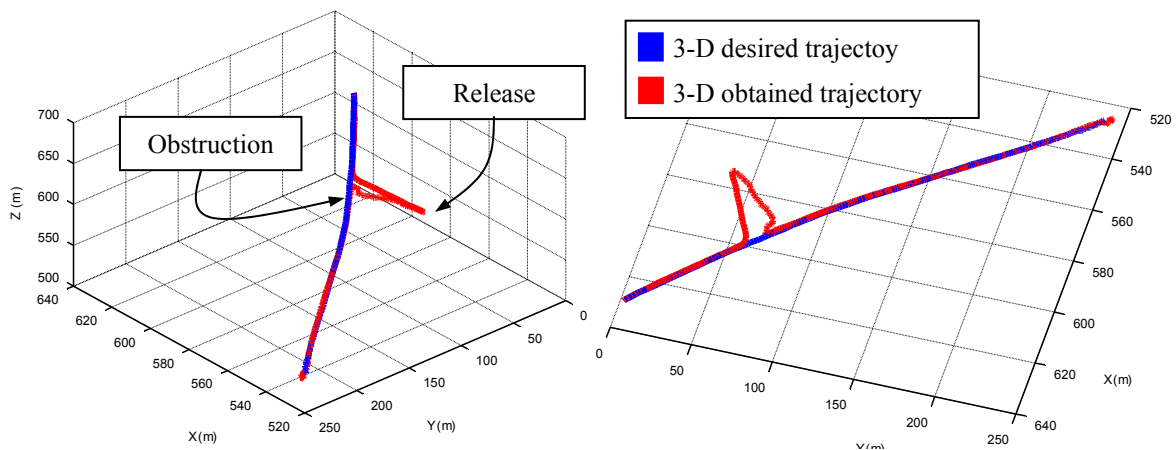


Figure 13. 3-D trajectory obtained using the time-independent method to track trajectories in the image based on virtual visual servoing.

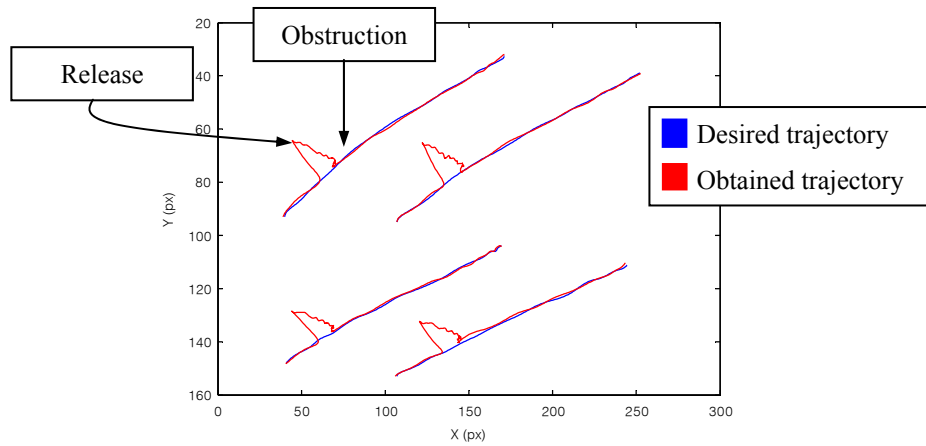


Figure 14. Evolution of the features in the image obtained using the time-independent method to track trajectories in the image based on virtual visual servoing.

These figures show the correct tracking of the desired path until the tracking is externally obstructed. Once the robot is released, the system pushes the robot back to the trajectory to complete it through the next unvisited reference.

Suppose now that a visual servoing system with temporal references (see equation (8)) is used for the tracking of the trajectory in the image. As in the previous mentioned case, within the tracking of the path, the robot is obstructed for 3 seconds. The obtained tracking is shown in Figure 15. The tracking system with time-dependent behaviour is not able to continue in the trajectory once the obstruction ends. This is due to the fact that when an unexpected event appears within the tracking (as an interaction with the environment) the temporal references are out of phase and become excessively restrictive. The system tries to maintain these temporal restrictions although not performing the correct tracking of the desired trajectory.

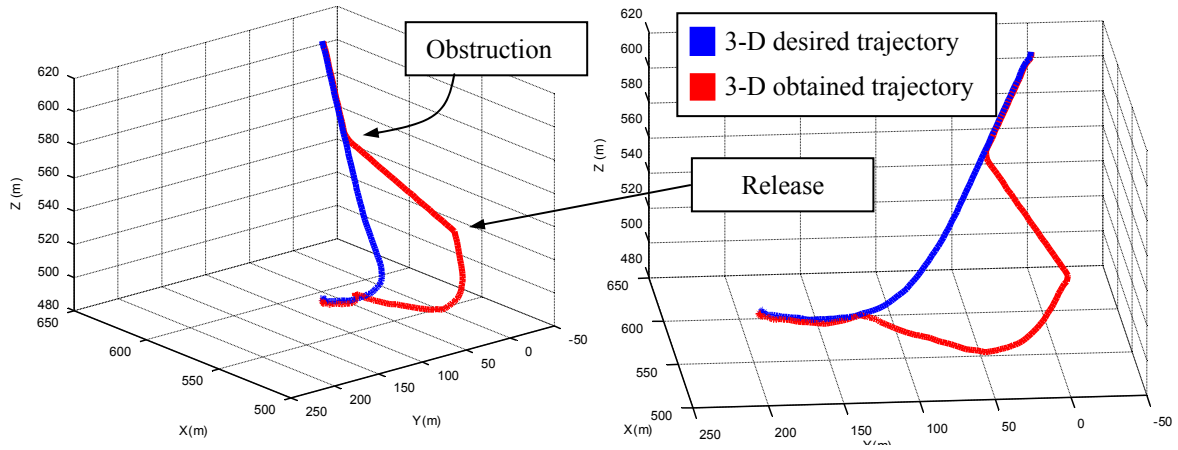


Figure 15. 3-D trajectory obtained with time-dependent systems (Test 1).

#### 5.4. Movement flow-based visual servoing versus the proposed method.

Movement flow-based visual servoing (Pomares & Torres, 2005) 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, to specify the velocity for the tracking is not possible. 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, an oscillating behaviour is obtained using this system.

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 16 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 close to  $|\mathbf{v}| = 100 \text{ mm/s}$ . In order to compare both methods, this desired velocity is used in the proposed

method. Figure 16 shows the trajectories followed by the four features in the image space. 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 17 demonstrates, the behaviour of the 3D trajectory is more accurate in the proposed method. This happens 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.

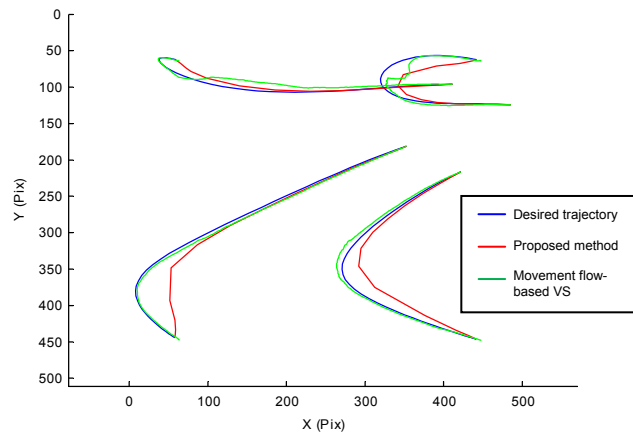


Figure 16. Image position of the features along the tracking using the proposed method and movement flow-based visual servoing

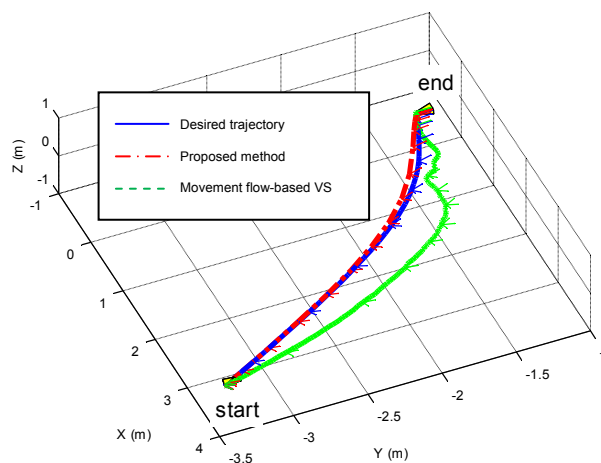
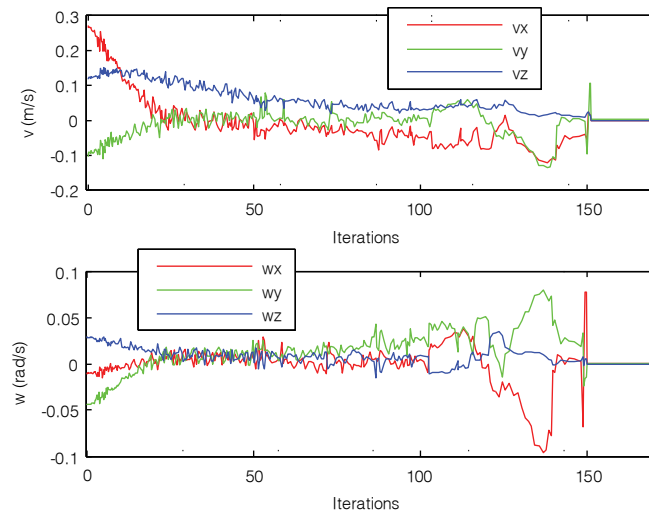


Figure 17. 3D position of the camera along the track using the proposed method and movement flow-based visual servoing

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 18 and Figure 19. In Figure 18 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 close to  $|\mathbf{v}|=100$  mm/s . Furthermore, this velocity can really be neither precisely nor directly controlled.

In Figure 19, 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 18. Linear and angular velocities of the camera along the tracking using movement flow-based visual servoing*

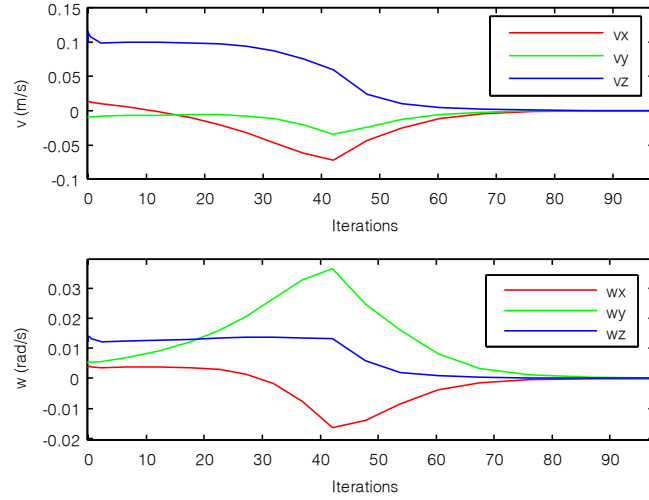


Figure 19. Linear and angular velocities of the camera along the tracking using the proposed method

Figure 19 shows that the velocity has been correctly controlled. Furthermore, as it can be seen in Figure 19, 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 18). Therefore, the proposed method to track trajectories based on visual servoing improves the behaviour of previous time-independent methods.

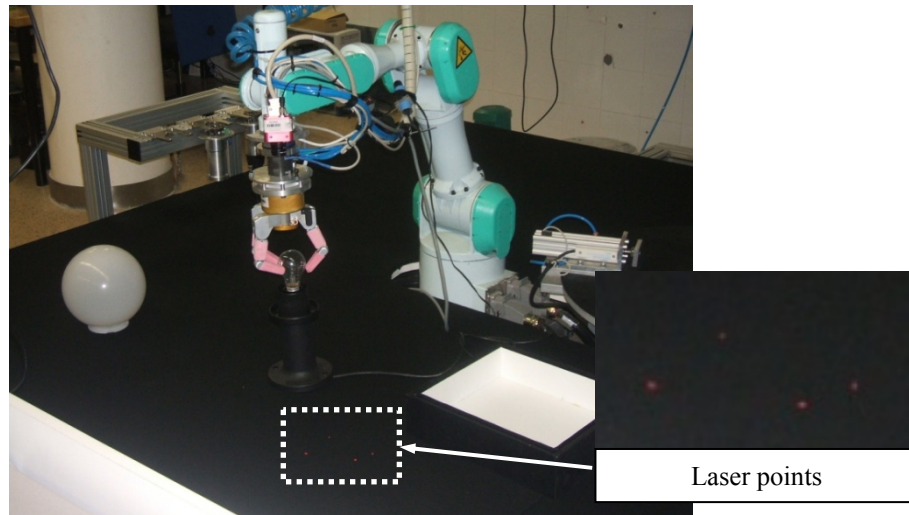
## 6. Results

In this section, the strategy to change automatically a bulb in a lamp is described. To do so, the robot is controlled using the visual path tracker described in Section 4. The use of the system proposed in this paper is necessary to obtain a global robust method to complete an automatic assembly and disassembly task. But, firstly, the experimental setup employed is described.

### 6.1. Experimental setup

The system architecture (see Figure 20) 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, which is also equipped with a force sensor (67M25A-I40 from JR3. Inc.). The camera is able to acquire and to process up to an image in 10 ms using an image resolution of 320x240. The robot is also equipped

with a Barrett hand. This paper does not focus in image processing issues. Therefore, the image trajectory is generated using four laser points projected on the table which will be the extracted features.



*Figure 20. Experimental setup*

## **6.2. Robust approach in the replacement of a non-working light bulb in a lamp**

The system described in this paper to track trajectories in the image is employed in the task of replacing a blown light bulb. The robot must be able to respond to any external circumstances to complete the task satisfactorily. For that reason, the image path tracker proposed here has been developed. When the robot must handle the lamp or the light bulb, it is necessary to control not only the position of the robot using a visual servoing approach, but also the forces measured at the end-effector. In this way, the robot is able to “see” and “feel”. The handle tasks are performed using the strain gauge measurements of the Barrett hand located at the end-effector. In these tasks, the concept of force-image interaction matrix defined in previous works (Garcia, Pomares, & Torres, 2007) has been used to modify the desired image trajectory. This force-image interaction matrix relates directly changes in the interaction forces measured at the end-effector with variations of the features in the image. When the robot performs large displacements, it must precisely track the desired path in the image in order to maintain a secure 3D path. The force feedback is not employed



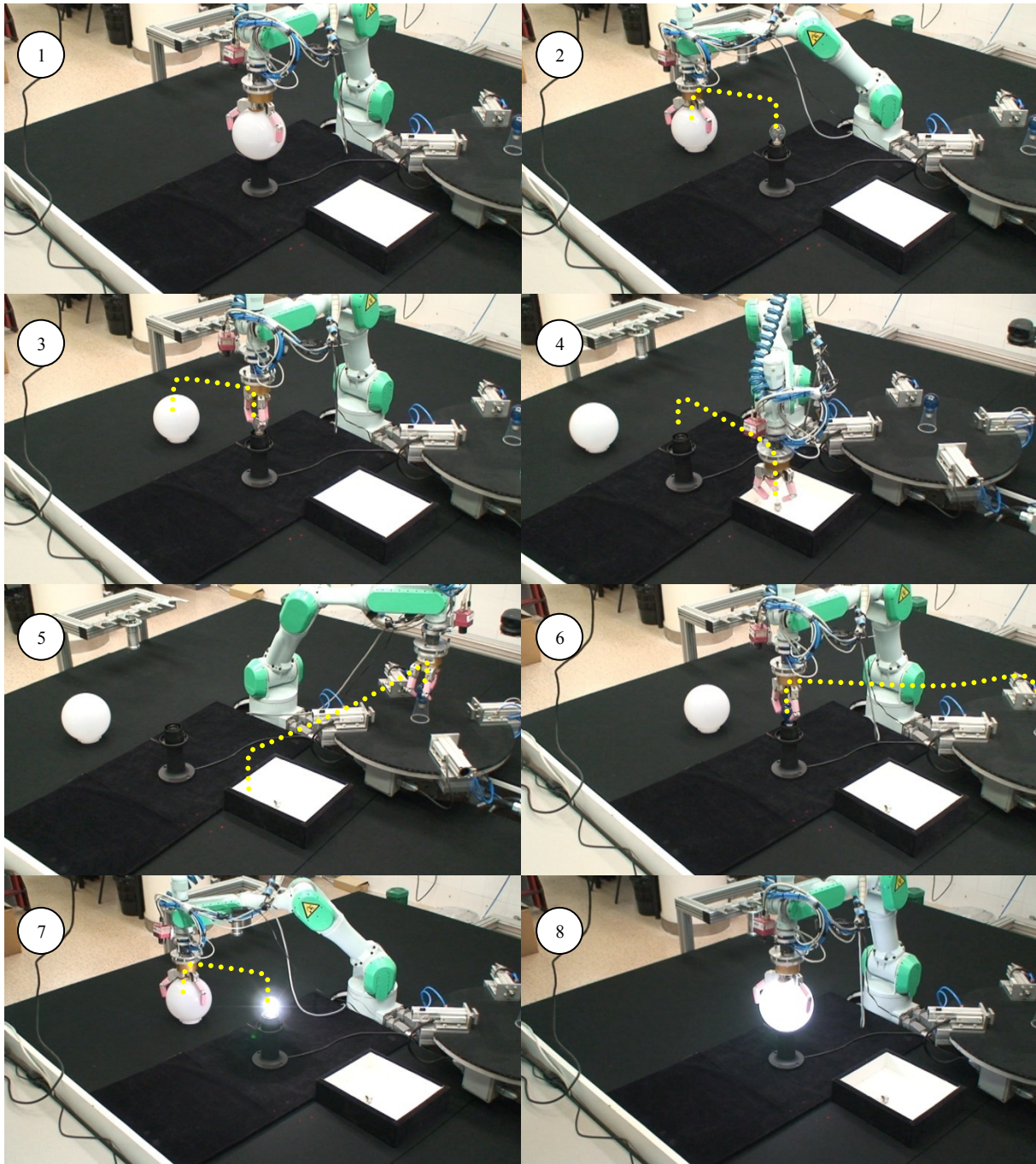
in these tasks in order to obtain an exact tracking of the path. The forces are still measured in order to detect any obstacle, but the desired path in the image is not changed when this obstruction occurs. The time-independent behaviour of the proposed path tracker permits the robot to go away the desired path maintaining the last point visited in the path as the next reference. Thus, when the interaction stops, the robot returns to the path at the same point it had left when the interaction began and the desired path is not modified.

The tracking system described in Section 4 is used in the different displacements required to complete the task. These displacements cannot be achieved using a classical image-based visual servoing approach because the 3D trajectory developed by the end-effector is not controlled (see Section 5.1). In the robot's workspace, different obstacles can avoid the completion of the task. The predefinition of secure 3D trajectories allows the robot to avoid these obstacles (see Figure 21). The proposed system is then employed to track these trajectories. Moreover, the time-independent behaviour of the proposed system allows the interaction with any object which enters into the secure zone, performing the correct tracking after the obstruction ends.

Image paths are stored in a previous phase. To do this, the robot is moved in a secure 3D path between two of its subsequent stages in the replacement task, storing the image trajectory during the movement. The image features are four laser points. These points have been artificially introduced because the lamp has no reference points. Furthermore, it is important to obtain a global method that does not depend on the lamp nor on its final location in the street.

Figure 21 shows the different stages to automatically complete the blown light bulb replacement task. Now, these stages are described:

1. Unscrewing the spherical light diffuser of the streetlamp. The light diffuser is grasped with a defined force measured by the Barrett hand. In this way, the correct grasping is assured as the three fingers hold firmly the object.



*Figure 21. Secure trajectories to be tracked by the time-independent method to track trajectories in the image based on virtual visual servoing*

2. The robot moves the spherical light diffuser to a temporal location. The path described by the robot is previously stored and is tracked using the proposed image path tracker.
3. In this stage the robot's arm returns to the lamp in order to unscrew the blown light bulb. As in the previous phase, the displacement is carried out by the time-independent image

path tracker. And, again, the light bulb is unscrewed by the Barrett hand, assuring that the defined desired force is used.

4. The non-working light bulb is placed on a deposit following a secure path.
5. Then the robot goes to grasp a new light bulb. Again, this displacement is developed with time-independent behaviour in order to follow a secure 3D path and to permit interactions with unexpected objects in this path.
6. Screwing in the new light bulb. The robot follows a secure path between the position in which it grasps the bulb and the lamp holder.
7. After this, the robot grasps the spherical light diffuser again employing the time-independent image path tracker proposed here.
8. Finally, the robot screws in the light diffuser in the streetlamp.

The different displacements described in these phases are carried out employing the proposed image path tracker. The time-independent behaviour of this approach is necessary to assure the correct tracking. Figure 22 shows the desired trajectory and the trajectory obtained for one of the laser points when the robot is obstructed during the task corresponding to the fourth stage shown in Figure 21. As it was described in Section 4, by using the proposed time-independent tracking method, the robot is able to recover the path and to follow a secure 3D path in order to avoid any undesired collision with any object (see Figure 22.a). Figure 22.b shows the result obtained when using a time-dependent approach and clearly demonstrates the necessity of this time-independent behaviour. It is crucial to obtain a robust system which permits the interaction and is able to complete the task satisfactorily.

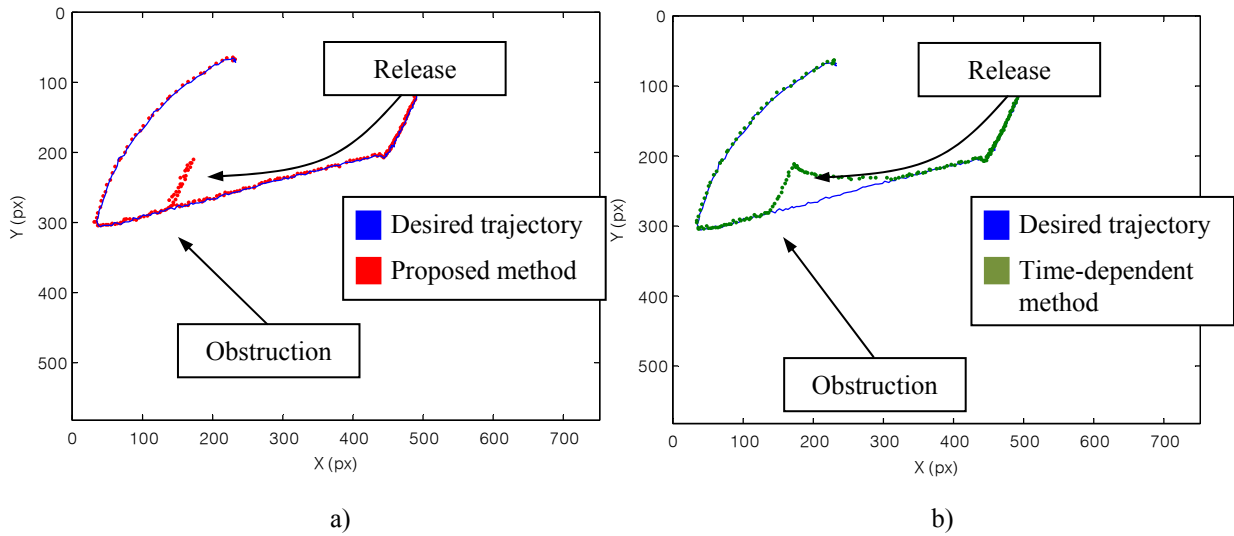


Figure 22. Image trajectory of one of the features using: a) the proposed time-independent image path tracker; and b) a time-dependent image path tracker.

Figure 23 shows the 3D trajectory obtained when the robot is obstructed during the task using the proposed time-independent image path tracker.

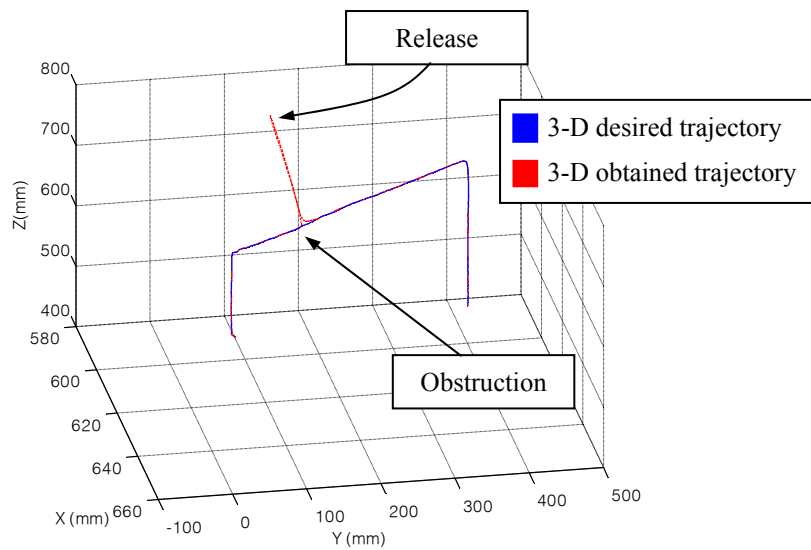
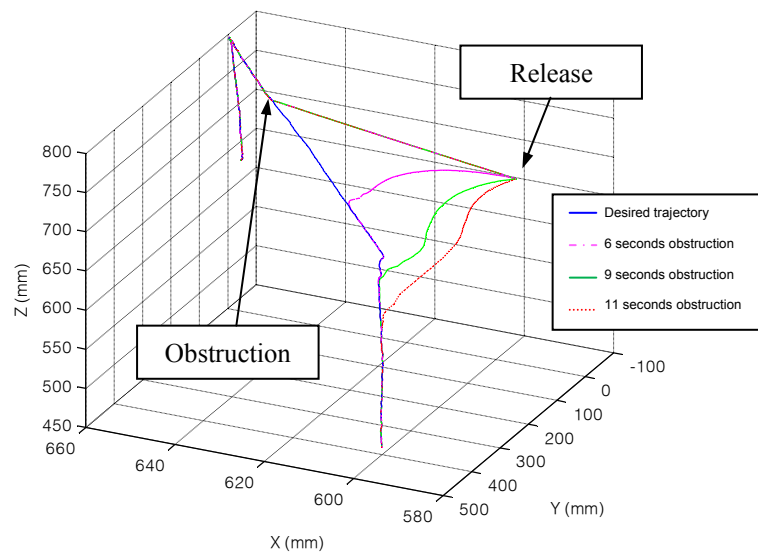


Figure 23. 3D trajectory of the end-effector of the robot with the proposed time-independent image path tracker.

Section 5.3 describes the difference between a time-independent method and a time-dependent one. An unexpected interaction in an automatic disassembly task can stop the robot for a long time. The desired behaviour in this case would be to continue with the trajectory once the obstruction

ends, independently of the length of the obstruction. Figure 24 shows how a time-dependent method becomes worse when the obstruction time increases. Comparing the results obtained with a time-dependent method (shown in Figure 24) and those obtained with the proposed method (see Figure 23), it can be observed that with the proposed method the system is able to complete satisfactorily the tracking of the path despite the obstruction. Figure 24 shows that a time-dependent method is not adequate to track completely the path because any interaction between the robot and an unexpected object will cause the loss of the path references.



*Figure 24. 3D trajectory of the end-effector of the robot with a time-dependent image path tracker for different obstruction times.*

## 7. Conclusions

An automatic robotic task requires a system able to react to unexpected situations inside the robotic cell. The application described in this paper uses a new method to track trajectories in the image using visual servoing. The tracking of a desired path allows the robot to perform secure trajectories in its workspace. Despite this, the robot can interact with a mobile object or a human. This interaction should not abort the task but only stop it. Once the obstruction ends the robot should be able to recover the path in the same point that it was tracking when this obstruction

began. The method proposed here to perform the tracking has a time-independent behaviour, guaranteeing the complete tracking of the path even in unexpected interactions as it has been remarked in the experimental results.

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 permits not only to control the velocity of the tracking, but also to obtain 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” and by the Valencian Community under Grant BPI06/015.

## **References**

- Chaumette, F. (1998). Potential problems of convergence in visual servoing. In *International Symposium on Mathematical Theory of Networks and Systems*, Padoue, Italy.
- Chaumette, F., & Hutchinson, S. (2006). Visual Servo Control, Part I: Basic Approaches. *IEEE Robotics and Automation Magazine*, 13(4), 82-90.
- Chesi, G., & Hashimoto, K. (2004). A simple technique for improving camera displacement estimation in eye-in-hand visual servoing. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(9), 1239-1242.
- Chesi, G., & Hung, Y. S. (2007). Global path-planning for constrained and optimal visual servoing. *IEEE Transactions on Robotics*, 23, 1050-1060.

- Chesi, G., Prattichizzo, D., & Vicino, A. (2007). Straight line path-planning in visual servoing. *Journal of Dynamic Systems Measurement and Control-Transactions of the ASME*, 129, 541-543.
- Chin, J.-H., Sun, Y.-H., & Cheng, Y.-M. (2008). Force computation and continuous path tracking for hydraulic parallel manipulators. *Control Engineering Practice*, 16(6), 697-709.
- Garcia, G.J., Pomares, J., & Torres, F. (2007). Robot guidance by estimating the force-image interaction matrix. *IFAC International Workshop on Intelligent Manufacturing Systems 2007*, Alicante, Spain.
- Hutchinson, S., Hager, G. D., & Corke, P. I. (1996) A tutorial on visual servo control. *IEEE Trans. Robotics and Automation*, 12(5), 651-670.
- Janschek, K., Tchernykh, V., & Dyblenk, S. (2007). Performance analysis of opto-mechatronic image stabilization for a compact space camera. *Control Engineering Practice*, 15(3), 333-347.
- Kinoshita, K., & Deguschi, K. (1994) Simultaneous determination of camera pose and intrinsic parameters by visual servoing. In *Proceedings of the 12th International Conference on Pattern Recognition*, (vol. A, pp. 285-289), Jerusalem, Israel.
- Malis, E. (2004). Visual servoing invariant to changes in camera-intrinsic parameters. *IEEE Transactions on Robotics and Automation*, 20, 72-81.
- Mansard, N., & Chaumette, F. (2007). Task Sequencing for High-Level Sensor-Based Control. *IEEE Transactions on Robotics*, 23(1), 60-72.
- Marchand, E., & Chaumette, F. (2001). A new formulation for non-linear camera calibration using VVS. *Publication Interne 1366, IRISA, Rennes, France*.
- Marchand, E., & Chaumette, F. (2002). Virtual visual servoing: a framework for real-time augmented reality. In *Proceedings of Eurographics 2002*, (vol.21, no.3, pp. 289-298), Saarebrücken, Germany.
- Mezouar, Y., & Chaumette, F. (2002). Path planning for robust image-based control. *IEEE Transactions on Robotics and Automation*, 18(4), 534-549.
- 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*, 35(1), 4-15.

- Pomares, J. Gil, P., Garcia, G. J., & Torres, F. (2006). 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, Czech Republic.
- Pomares, J., Chaumette, F., & Torres, F. (2007). Adaptive visual servoing by simultaneous camera calibration. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, (vol. 1-10, pp. 2811-2816), Rome, Italia.
- Schramm, F., & Morel, G. (2006). Ensuring visibility in calibration-free path planning for image-based visual servoing. *IEEE Transactions on Robotics*, 22(4), 848-854.
- Shoemake, K. (1985). Animating Rotation with Quaternion Curves. *Computer Graphics* 19(3), 245-254.
- Soueres, P., Hamel, T., & Cadenat, V. (1998). A path following controller for wheeled robots which allows to avoid obstacles during transition phase. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, (vol.2, pp.1269-1274), Pasadena, USA.
- Tsai, R. Y. (1987). A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4), 323-344.
- Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. *International Conference on Computer Vision*, (pp. 666-673), Corfu, Greece.