Workshop on New Trends in Image-based Robot Servoing, IROS'97, pp. 45-52, Grenoble, France, September 1997.

2D 1/2 visual servoing with respect to a planar object

François Chaumette Ezio Malis Sylvie Boudet

IRISA / INRIA Rennes
Campus de Beaulieu
35 042 Rennes, France

EDF-DER
6 Quai Watier
78 401 Chatou, France

E-mail: {chaumett, emalis}@irisa.fr, sylvie.boudet@edfgdf.fr

Abstract

In this paper, we propose a new vision-based robot control approach which is halfway between the classical position-based and image-based visual servoings. It allows to avoid their respective disadvantages. For a planar object, the homography between the feature points extracted from two images (corresponding to the current and desired camera poses) is computed at each iteration of the control law. From this homography, an approximate partial-pose, where the translational term is known only up to a scale factor, is deduced. Using parameters of this partial pose and image features, it is possible to design a closed-loop control law controlling the six camera d.o.f. Contrarily to the position-based visual servoing, our scheme does not need any geometric 3D model of the object. Furthermore and contrarily to the image-based visual servoing, our approach ensures the convergence of the control law in all the task space.

1 Introduction

Vision-based robot control using an eye-in-hand system is generally performed using two different approaches [11, 8, 9]: *position-based* and *image-based* control systems. In a *position-based* control system, the control error function is computed in the 3D Cartesian space [10] (for this reason this approach can be called *3D visual servoing*). The pose of the target with respect to the camera, which describes its 3D position and 3D orientation, is estimated from image features corresponding to the perspective projection of the target in the image. Numerous methods exist to recover the pose of an object (see [4] for example). They are all based on the knowledge of a perfect geometric model of the object and necessitate a calibrated camera to obtain unbiased results. The main advantage of 3D visual servoing is that it controls the camera trajectory in the Cartesian space, which allows to easily combine the visual positioning task with obstacles avoidance and/or joint limits and singularities avoidance. On the other hand, in an *image-based* control system, the control error function is computed in the 2D image space (for this reason this approach can be called *2D visual servoing*). This local approach is based on the use of an image Jacobian (also called interaction matrix [5]) and the control laws provide at each iteration the camera velocity for minimizing the observed error in the image. This approach is known to be very robust with respect to camera and robot calibration errors [6]. However, the convergence is ensured only in a region (quite impossible to determine) around the desired position. Furthermore, the Cartesian trajectory of the camera is uncontrolled, and some image features may get out of the camera field of view during the servoing, which generally gives rise to its failure.

The purpose of this paper is to design a new visual servoing system which combines the advantages of 2D and 3D visual servoings and avoids their drawbacks. We point out our attention to one of the typical applications of visual servoing: positioning a camera mounted on a robot end-effector relative to a target, for a grasping task for instance. The positioning task is divided into two steps. In the first off-line learning step, the camera is moved to its desired position. The image of the target corresponding to this position is acquired and the extracted desired features are stored. In the second on-line step, after the camera and/or the target have been moved, the camera is commanded so that the current features reach their desired position in the image. In such application, the target geometry is not always precisely known. Furthermore, the convergence has to be obtained in all the task space.

Our approach can be called *2D 1/2 visual servoing* since the control error function is computed in part in the 3D Cartesian space and in part in the 2D image space. More precisely, the homography between the feature points extracted from two images (corresponding to the current and desired camera poses) is computed at each iteration. From the homography, the rotation of the camera between the two views is estimated. Consequently, the rotational control loop can be decoupled from the translational one. A such decoupled system allows to obtain the convergence in all the task space. It must be emphasized that the use of an homography does not need neither a 3D model of the target (the method we actually use to estimate the homography is however only valid for planar objects) nor a precisely calibrated camera. As far as the camera translational displacement is concerned, it can only be estimated up to a scale factor. It is therefore controlled using 2D image features and the ratio, obtained from the homography, of the unknown desired and current distances between the camera and the target. Finally, in spite of the partial-pose estimation, experiments show that the approach is robust to calibration errors.

The paper is organized as follows. In Section 2 and Section 3 we briefly recall 3D and 2D visual servoings respectively. In Section 4 we present the 2D 1/2 visual servoing approach. The experimental results are given in Section 5.

2 3D Visual Servoing

Let F_0 be the coordinate frame attached to the target, F_1 and F_2 be the coordinate frames attached to the camera in its desired and current position respectively (see Figure 1).

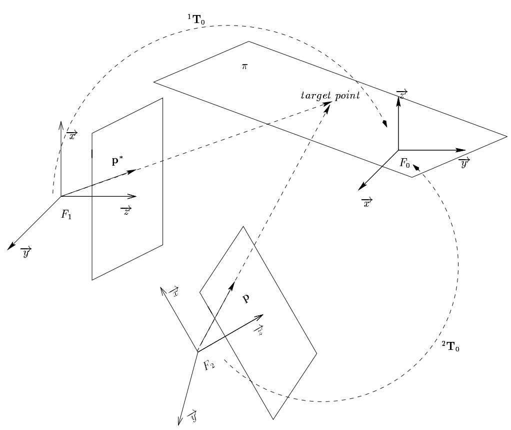


Figure 1: Modelisation of camera displacement for 3D visual servoing

Knowing the coordinates, expressed in F_0, of at least four points of the target [4], it is possible from their measure in the image to compute the desired camera pose (represented in Figure 1 by the transformation matrix ${}^1\mathbf{T}_0$) and the current camera pose (represented in Figure 1 by the transformation matrix ${}^2\mathbf{T}_0$). The camera displacement ${}^2\mathbf{T}_1$ to reach the desired position is thus easily obtained (${}^2\mathbf{T}_1 ={}^2 \mathbf{T}_0 \left({}^1\mathbf{T}_0\right)^{-1}$), and can be performed either in open loop or, more robustly, in closed loop by computing ${}^2\mathbf{T}_0$ at each iteration of the control law. Finally, the control law has to be designed in order that the image features used in the pose estimation always appear in the camera field of view. This can not be guaranteed if the camera or the robot are only coarse-calibrated. Let us also note that if the camera is not accurately calibrated, or if the 3D model of the considered object is not perfectly known, the estimated current and desired camera poses will be biased. However, this is not a drawback if a closed-loop scheme is performed.

The corresponding block diagram of the 3D visual servoing approach is given in Figure 2.

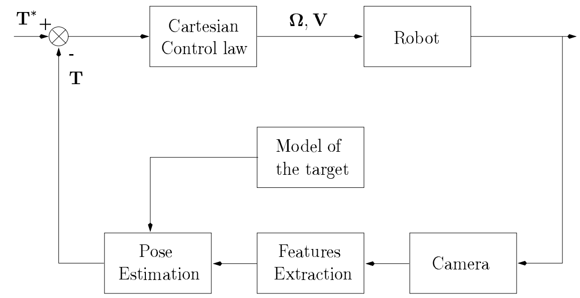


Figure 2: Block diagram of the 3D visual servoing

3 2D Visual Servoing

The control error function is now expressed directly in the 2D image space (see Figure 3).

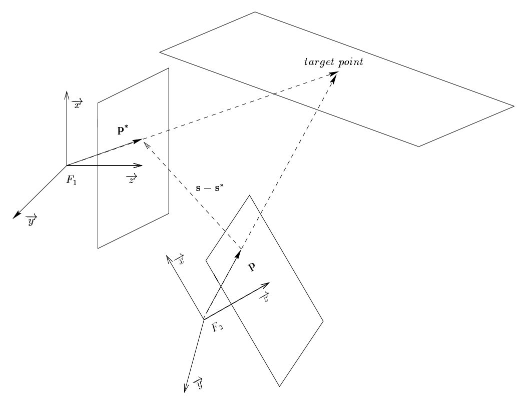


Figure 3: Modelisation of camera displacement for 2D visual servoing

Let $\mathbf{s}$ be the current value of visual features observed by the camera and $\mathbf{s}^*$ be the desired value of $\mathbf{s}$ to be reached in the image. The time variation of $\mathbf{s}$ is related to camera velocity by [5]:

$$\dot{\mathbf{s}} = \mathbf{L}(\mathbf{s}, Z)\mathbf{T} \tag{1}$$

where $\mathbf{T} = \left[\begin{array}{cc} \mathbf{V} & \mathbf{\Omega} \end{array} \right]$ is the camera velocity screw and $\mathbf{L}(\mathbf{s}, Z)$ is the interaction matrix related to $\mathbf{s}$. For example, if the chosen features are the coordinates $\mathbf{s} = \left[\begin{array}{cc} u & v \end{array} \right]^T = \left[\begin{array}{cc} X/Z & Y/Z \end{array} \right]^T$ in the image of a 3D point $\mathbf{P}$ of coordinates $\left[\begin{array}{ccc} X & Y & Z \end{array} \right]^T$ in the camera frame, the interaction matrix related to u and v is given by:

$$\mathbf{L}(\mathbf{s}, Z) = \begin{bmatrix} \frac{1}{Z}\mathbf{L}_v(\mathbf{s}) & \mathbf{L}_\omega(\mathbf{s}) \end{bmatrix} \tag{2}$$

where:

$$\mathbf{L}_v(\mathbf{s}) = \begin{bmatrix} -1 & 0 & u \\ 0 & -1 & v \end{bmatrix} \tag{3}$$

$$\mathbf{L}_\omega(\mathbf{s}) = \begin{bmatrix} uv & -(1+u^2) & v \\ (1+v^2) & -uv & -u \end{bmatrix} \tag{4}$$

The interaction matrix for more complex image features can be found in [5]. The vision-based task $\mathbf{e}$ (to be regulated to 0), corresponding to the regulation of $\mathbf{s}$ to $\mathbf{s}^*$, is defined by:

$$\mathbf{e} = \mathbf{C}(\mathbf{s} - \mathbf{s}^*) \tag{5}$$

where $\mathbf{C}$ is a matrix which has to be selected such that $\mathbf{CL}(\mathbf{s}, Z) > 0$ in order to ensure the convergence of the control law. The optimal choice seems to consider $\mathbf{C}$ as the pseudo-inverse $\mathbf{L}(\mathbf{s}, Z)^+$ of the interaction matrix. The matrix $\mathbf{C}$ thus depends on the depth Z of each target point used in the visual servoing. An estimation of the depth can be obtained using, as in 3D visual servoing, a pose determination algorithm (if a 3D target model is available), or using a structure from known motion algorithm (if the camera motion can be measured). However, using this choice for $\mathbf{C}$ may lead the system to near, or even reach, a singularity of the interaction matrix. Furthermore, the convergence may also not be attained due to local minima reached because of the computation by the control law of unrealizable motions in the image [2].

Another choice is to consider $\mathbf{C}$ as a constant matrix equal to $\mathbf{L}(\mathbf{s}^*, Z^*)^+$, the pseudo-inverse of the interaction matrix computed for $\mathbf{s} = \mathbf{s}^*$ and $Z = Z^*$, where Z^* is an approximate value of Z at the desired camera position. In this simple case, the condition for convergence is however only satisfied in the neighborhood of the desired position, which means that the convergence may not be ensured if the initial camera position is too far away from the desired one.

If an exponential convergence is desired ($\dot{\mathbf{e}} = -\lambda\mathbf{e}$ where λ is a positive scalar), a simple control law is given by [5]:

$$\mathbf{T} = -\lambda\mathbf{e} = -\lambda\mathbf{C}(\mathbf{s} - \mathbf{s}^*) \tag{6}$$

The block diagram of the 2D visual servoing approach is given in Figure 4.

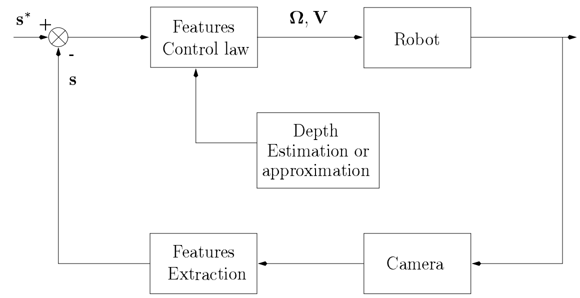


Figure 4: Block diagram of the 2D visual servoing

4 2D 1/2 Visual Servoing

If n 3D points $\mathbf{P}_j$ on a planar target are used, it is well known that each image point $\mathbf{p}_j$, of coordinates $\begin{bmatrix} u & v & 1 \end{bmatrix}^T$ in the current camera frame F_2, is related to the corresponding image point $\mathbf{p}_j^*$, of coordinates $\begin{bmatrix} u^* & v^* & 1 \end{bmatrix}^T$ in the desired camera frame F_1, by an homography such that [7]:

$$\alpha_j \mathbf{p}_j = \mathbf{H}\mathbf{p}_j^* \qquad \{j = 0, 1, 2, ..., n-1\} \tag{7}$$

where α_j is a positive scalar and $\mathbf{H}$ is a (3×3) homography matrix. More precisely, $\mathbf{H}$ can be written [7]:

$$\mathbf{H} = \mathbf{R} + \mathbf{n}^* \frac{\mathbf{t}^T}{d^*} \tag{8}$$

where $\mathbf{R}$ and $\mathbf{t}$ are the rotational matrix and the translational vector between the current camera frame F_2 and the desired camera frame F_1 respectively (see Figure 5), $\mathbf{n}^* = \begin{bmatrix} n_x^* & n_y^* & n_z^* \end{bmatrix}^T$ is the unit vector normal to the target plane π expressed in F_1 and d^* is the distance between the origin of F_1 and π.

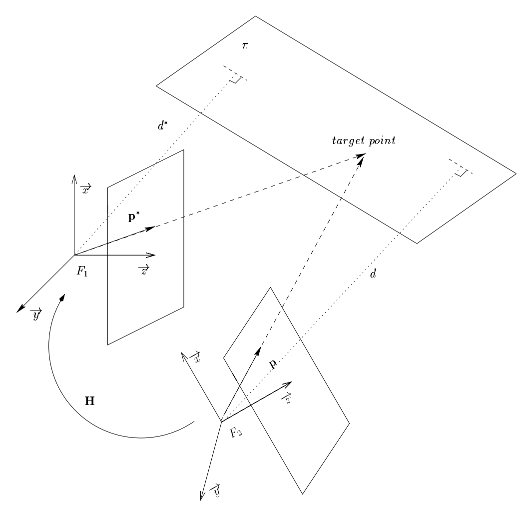


Figure 5: Modelisation of camera displacement for 2D 1/2 visual servoing

Let us remark that $\mathbf{H}$ is defined up to a scalar factor, therefore one of the unknown α_j can be set to 1 without loss of generality (for example $\alpha_0 = 1$). Equation (7) is available for each feature point in the image, thus, for n points, we have to solve a linear system of $3n$ equations and $n+8$ unknowns: that is $n-1$ unknowns α_j ($\{j=1,2,...,n-1\}$ since $\alpha_0 = 1$), and 9 unknown elements of $\mathbf{H}$. A minimum of four points is needed to perform a linear estimation. Let us emphasize that the estimation of $\mathbf{H}$ does not necessitate the knowledge of the 3D position of the target points on plane π, which makes unnecessary a 3D model of the target.

After matrix $\mathbf{H}$ is computed, $\mathbf{R}$, $\mathbf{t}/d^*$ and $\mathbf{n}^*$ can be estimated [7]. Unfortunately, in the most general case we have two different solutions. The indetermination is eliminated by choosing the solution which is such that the vector $\mathbf{n}^*$ is as co-linear as possible with the desired orientation of the camera optical axis. Let us note that the translational vector $\mathbf{t}$ is estimated only up to a scale factor since the desired distance d^* is unknown. 3D visual servoing can thus not be employed. It is possible to design a control law such that $\mathbf{R}$ and $\mathbf{t}/d^*$ respectively have to reach the identity matrix $\mathbf{I}$ and $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ (which thus implies the achievement of the positioning task). However, such a control law does not ensure that the considered object will always remain in the camera field of view since it is only based on 3D estimated parameters. Getting out of the image may thus occur in the presence of important errors in the intrinsic parameters of the camera or in the robot Jacobian.

We have therefore preferred another more robust solution. The block diagram of the 2D 1/2 visual servoing approach is given in Figure 6. We now describe its differents parts. Our control scheme is based on the 3D estimated rotation between F_2 and F_1 (which has to reach the identity matrix). We also use the ratio r which controls the depth between the camera and the target (and which has to reach the desired value $r^* = 1$). Indeed, the distances d and d^* are unknown, but the ratio $r = d/d^*$ can easily be estimated using $\mathbf{R}$, $\mathbf{t}/d^*$ and $\mathbf{n}^*$. We remark that the current normal vector $\mathbf{n}$ to the plane π, expressed in F_2, and the current distance d between frame F_2 and π can be written in function of vector $\mathbf{n}^*$ and distance d^*:

$$\mathbf{n} = \mathbf{R}\mathbf{n}^* \tag{9}$$

$$d = d^* + \mathbf{n}^T\mathbf{t} \tag{10}$$

From equations (9) and (10) we get:

$$r = \frac{d}{d^*} = 1 + \mathbf{n}^T\frac{\mathbf{t}}{d^*} = 1 + \mathbf{n}^{*T}\mathbf{R}^T\frac{\mathbf{t}}{d^*} \tag{11}$$

Finally, in order to control the 6 camera d.o.f and to maintain the target in the camera field of view, we also introduce the use of two independent visual features, such as the image coordinates of a target point, which can be controlled by using classical 2D visual servoing.

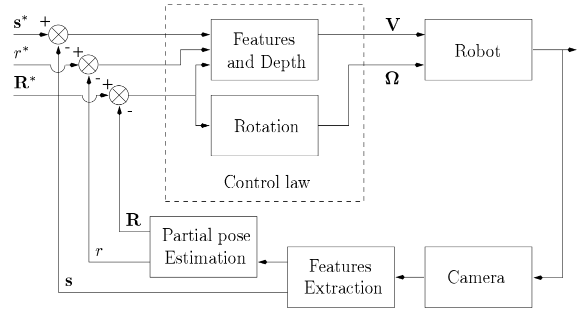


Figure 6: Block diagram of the 2D 1/2 visual servoing

We first consider the control of the camera orientation. Let $\mathbf{u}$ be the rotation axis and θ the rotation angle obtained from the previous partial-pose estimation. The rotational velocity of the camera $\mathbf{\Omega}$ can be expressed in function of the angular velocity $\dot{\theta}$ around the axis of rotation $\mathbf{u}$:

$$\mathbf{u}\,\dot{\theta} = \mathbf{\Omega} \tag{12}$$

The translational degrees of freedom are used to maintain the target in the image and to control the ratio r between the current distance d and the desired distance d^*. A point $\mathbf{p}^*$ is chosen in the desired image, such as for example the nearest from the image center. As done for the 2D visual servoing, the chosen features are its coordinates $\mathbf{s} = \begin{bmatrix} u & v \end{bmatrix}^T$ in the image. By using equations (1) and (2), the relationship between the time variation of the chosen features and the velocity of the camera is:

$$\dot{\mathbf{s}} = \frac{1}{Z}\mathbf{L}_v(\mathbf{s})\mathbf{V} + \mathbf{L}_\omega(\mathbf{s})\mathbf{\Omega} \tag{13}$$

where Z is the depth of the corresponding 3D point. Since the 3D point belongs to the target plane π, Z can be written as:

$$Z = \frac{d}{\mathbf{n}^T\mathbf{p}} \tag{14}$$

and from equation (11), we get:

$$\frac{1}{Z} = \frac{\mathbf{n}^T\mathbf{p}}{rd^*} = \frac{\gamma}{d^*} \tag{15}$$

where $\gamma = \frac{\mathbf{n}^T\mathbf{p}}{r}$ can be computed from image measurement and the partial-pose algorithm. Furthermore, by derivating equation (11) with respect to time, we get:

$$\begin{aligned} \dot{r} &= \frac{\mathbf{n}^{*T}}{d^*}(\mathbf{R}^T\dot{\mathbf{t}} + \dot{\mathbf{R}}^T\mathbf{t}) && (16)\\ &= \frac{\mathbf{n}^{*T}}{d^*}\mathbf{R}^T(\dot{\mathbf{t}} + \mathbf{R}\dot{\mathbf{R}}^T\mathbf{t}) && (17)\\ &= \frac{\mathbf{n}^T}{d^*}(\dot{\mathbf{t}} + \Omega \times \mathbf{t}) && (18)\\ &= -\frac{\mathbf{n}^T}{d^*}\mathbf{V} && (19) \end{aligned}$$

since, from the fundamental kinematics equation, we have $\dot{\mathbf{t}} = -\mathbf{V} - \Omega \times \mathbf{t}$. Combining equation 19 to equation (12) and equation (13), we get the following system:

$$\begin{bmatrix} \dot{\mathbf{s}} \\ \dot{r} \\ \mathbf{u}\dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{d^*}\mathbf{M}_v & \mathbf{M}_\omega \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{\Omega} \end{bmatrix} \tag{20}$$

where:

$$\mathbf{M}_v = \begin{bmatrix} -\gamma & 0 & \gamma u \\ 0 & -\gamma & \gamma v \\ -n_x & -n_y & -n_z \end{bmatrix} \tag{21}$$

$$\mathbf{M}_\omega = \begin{bmatrix} uv & -(1+u^2) & v \\ (1+v^2) & uv & -u \\ 0 & 0 & 0 \end{bmatrix} \tag{22}$$

Equation (20) is similar to equation (1). The task function approach [5] can thus be applied. If we desire an exponential convergence of s towards s^*, of r towards $r^* = 1$ and $\mathbf{u}\theta$ towards $\mathbf{u}\theta^* = 0$ (with decreasing velocity λ), we obtain:

$$\begin{bmatrix} \dot{\mathbf{s}} \\ \dot{r} \\ \mathbf{u}\dot{\theta} \end{bmatrix} = -\lambda \begin{bmatrix} \mathbf{s} - \mathbf{s}^* \\ r - 1 \\ \mathbf{u}\theta \end{bmatrix} \tag{23}$$

From equations (20) and (23), the control law is given by:

$$\begin{bmatrix} \mathbf{V} \\ \mathbf{\Omega} \end{bmatrix} = -\lambda \begin{bmatrix} d^*\mathbf{M}_v^{-1} & -d^*\mathbf{M}_v^{-1}\mathbf{M}_\omega \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{s} - \mathbf{s}^* \\ r - 1 \\ \mathbf{u}\theta \end{bmatrix} \tag{24}$$

where $\mathbf{M}_v^{-1}$, the inverse matrix of $\mathbf{M}_v$, is given by:

$$\mathbf{M}_v^{-1} = \frac{-1}{\mathbf{n}^T\mathbf{p}} \begin{bmatrix} \frac{n_z + n_y v}{\gamma} & -\frac{n_y u}{\gamma} & \frac{u}{\gamma} \\ -\frac{n_x v}{\gamma} & \frac{n_z + n_x u}{\gamma} & \frac{v}{\gamma} \\ -n_x & -n_y & 1 \end{bmatrix} \tag{25}$$

The determinant of matrix $\mathbf{M}_v$ is $-\gamma^2(\mathbf{n}^T\mathbf{p})$. This determinant is null only if the optical axis of the camera belongs to plane π (in this singular case, all the image points are collinear), which ensures the stability of the system in all the task space (more precisely, the half space in front of the target plane) if θ and $\mathbf{n}$ are correctly estimated.

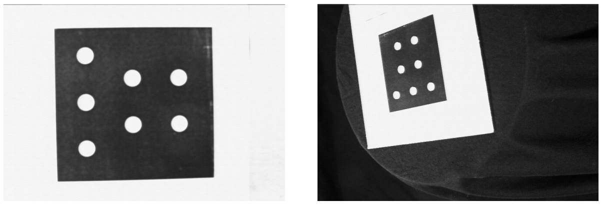
We note that the positive scalar γ only influences the time-to-convergence of the translational control loop. Indeed, if γ is estimated as the positive scalar $\overline{\gamma}$ and vector $\mathbf{n}$ is well-estimated, the matrix $\mathbf{M}_v^{-1}(\overline{\gamma})\mathbf{M}_v(\gamma) = \frac{\overline{\gamma}}{\gamma}\mathbf{I}$ is always definite positive. We can also note that the camera translational velocity is proportional to the desired distance d^* between F_2 and π. An approximate value has thus to be chosen during the off-line learning stage. However, this value has not to be precisely determined (by hand in the following experiments) since it does not influence the stability of the system, but only the time-to-convergence of the translational velocity and the amplitude of the possible tracking error due to a wrong compensation of the rotational motion. As far as the tracking error is concerned, it is proportional to the rotational velocity $\mathbf{\Omega}$ and thus disappears when the camera is correctly oriented.

5 Experimental results

The control law has been tested on a seven d.o.f. industrial robot MITSUBISHI PA10 (at EDF DER Chatou) and a six d.o.f. cartesian robot AFMA (at IRISA - see Figure 7). The camera is mounted on the end-effector of the robot. The target is a black board with seven white marks (see Figure 8). The extracted visual features are the image coordinates of the center of gravity of each mark. As far as camera calibration is concerned, we have used the pixel and focal lengths given by the constructor in order to compute the image coordinates u and v from their measured values in the image. The center of the image has been used for the projection of the optical axis in the image. The images corresponding to the desired and initial camera position are plotted in Figure 8a and 8b respectively.

Figure 7: Experimental cell at IRISA

Classical 2D visual servoing has first been tested, but without success due to the too much important



(a) (b)

Figure 8: Images of the target for the desired (a) and the initial (b) camera position

displacement between the initial and desired positions. Matrix $\mathbf{C}$ of equation (5) was chosen always equal to $\mathbf{L}^{+}(\mathbf{s}^*, Z^*)$ because Z cannot be calculated without a geometric model of the target. The errors on the coordinates of the center of gravity of each mark are given in Figure 9. The corresponding trajectory in the image is given in Figure 10.

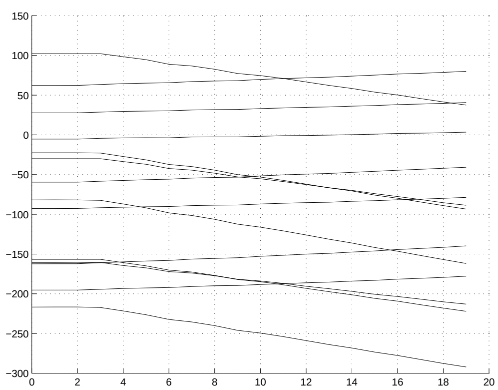


Figure 9: Error on the coordinates of the image features (pixels) versus iteration number using 2D visual servoing

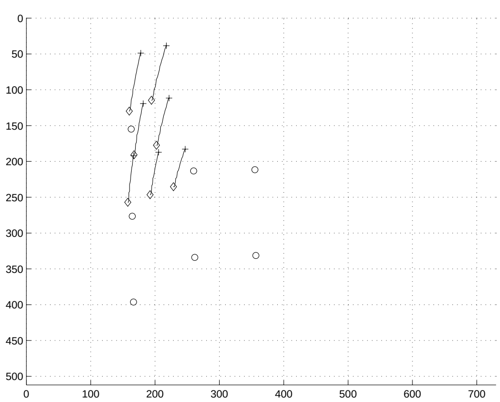


Figure 10: Trajectory of target points in the image using 2D visual servoing

On the obtained plots, we can note the divergence of the visual features, which induces the getting out of the target outside the camera field of view, and thus the failure of the 2D visual servoing. The corresponding translational and rotational velocity are given together in Figure 11.

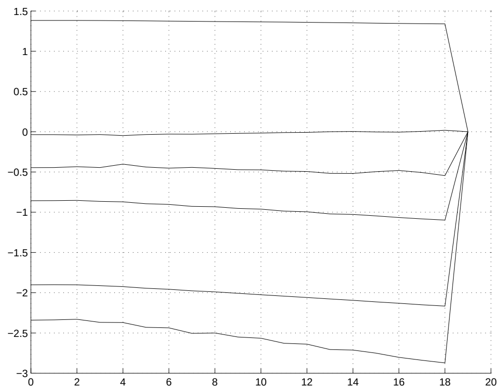


Figure 11: Translational velocity V (cm/s), rotational velocity Ω (dg/s) versus iteration number using 2D visual servoing

We now present the results obtained using our new 2D 1/2 visual servoing scheme. In the presented experiment, d^* is set to 20 cm. From the estimated homography, we get a partial estimation of the camera displacement. For example, the estimated rotational displacement, using the initial and desired images, was $\overline{r}_x$ = 29.6 dg, $\overline{r}_y$ = -32.5 dg, $\overline{r}_z$ = 95.9 dg (while the real displacement was r_x= 28.1 dg, r_y = -34.0 dg, r_z = 96.1 dg). Similarly, the estimated direction of translation was $\frac{\overline{t}_x}{\|\mathbf{t}\|} = -0.19$, $\frac{\overline{t}_y}{\|\mathbf{t}\|} = 0.95$, $\frac{\overline{t}_z}{\|\mathbf{t}\|} = -0.23$ (while the real displacement was t_x = -3.8 cm, t_y = 23.6 cm, t_z = -5.9 cm, which implies $\frac{t_x}{\|\mathbf{t}\|} = -0.15$, $\frac{t_y}{\|\mathbf{t}\|} = 0.95$, $\frac{t_z}{\|\mathbf{t}\|} = -0.23$. The used algorithm is thus quite precise (maximal rotational error is around 2 dg, as well as the angle error on the direction of translation) in despite of the coarse calibration which has been used.

The exponential decreasing of the estimated rotation, of ratio $r = d/d^*$ and of the coordinates u and v of the target point in the image are given in Figure 12, Figure 13 and Figure 14 respectively.

As can be seen on the plots, the obtained results are particularly stable and robust. In this experiment, λ is set to 0.1. A smaller value just implies a slower time-to-convergence. Higher values could also be chosen. However, the system (whose rate is 10 Hz) becomes unstable when $\lambda > 2.5$.

A good approximation of d^* reduces the amplitude of the tracking error due to the rotational movement. A such small perturbation can be observed in Figure 14 from iteration 120 to 250 where the tracking error is approximately 3 pixels. Let us note that the tracking

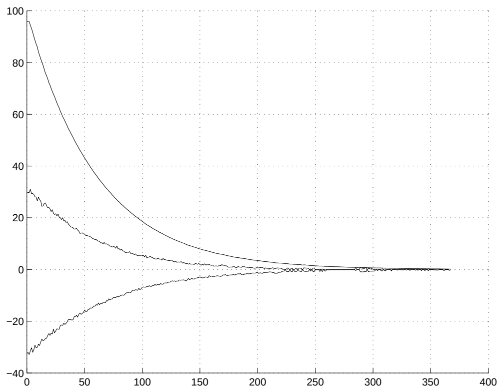


Figure 12: Rotation $\mathbf{u}\theta$ (dg) versus iteration number

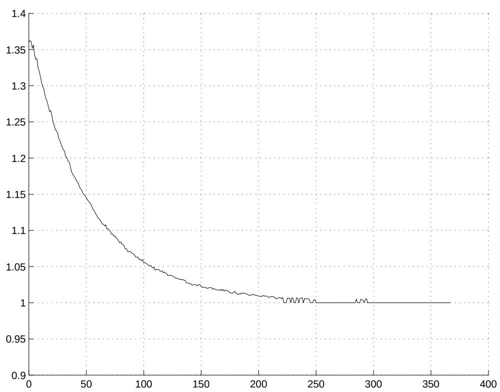


Figure 13: Ratio $r = d/d^*$ versus iteration number

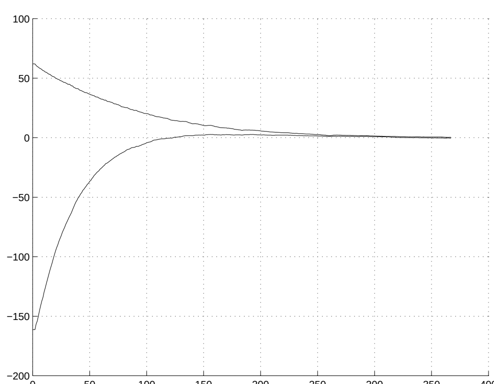


Figure 14: Error on the coordinates of the target center of gravity (pixels) versus iteration number

error completely disappears (after iteration 250) when the rotational motion has no more influence.

The outputs of the control law, $\mathbf{V}$ and $\mathbf{\Omega}$ are given in Figure 15.

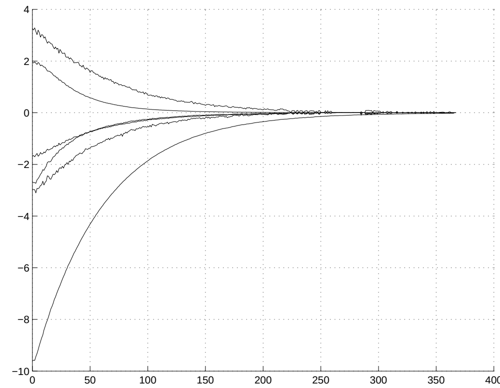


Figure 15: Translational velocity V (cm/s), rotational velocity Ω (dg/s) versus iteration number using 2D 1/2 visual servoing

Once again, we can note the stability and the robustness of the control law. Finally, the error on the image coordinates of each target point is given in Figure 16 and the corresponding trajectory in the image is given in Figure 17.

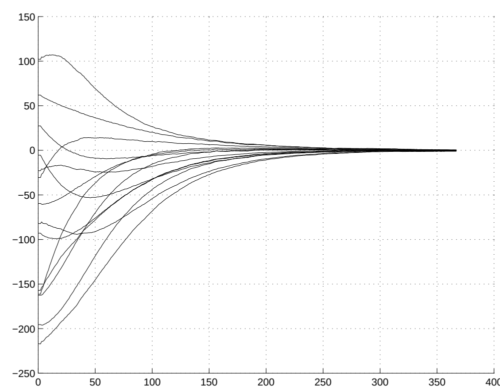


Figure 16: Error on the coordinates of the image features (pixels) versus iteration number using 2D 1/2 visual servoing

We can note the convergence of the coordinates to their desired values (the control scheme is stopped when maximal error is less than 0.5 pixels), which demonstrates the correct achievement of the positioning task. Let us recall that these image coordinates are only used to compute the homography between initial and final camera positions.

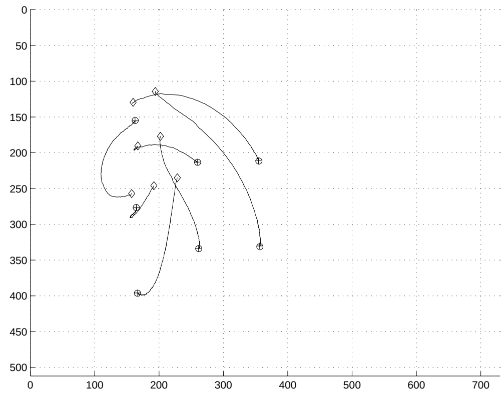


Figure 17: Trajectory of target points in the image using 2D 1/2 visual servoing

6 Conclusion

In this paper, we have proposed a new approach to vision-based robot control. Contrarily to the pose-based approach, the proposed control law does not need any geometric model of the target. Furthermore, the domain of convergence for the 2D 1/2 visual servoing (the half space in front of the target) is larger than for the 2D visual servoing. Finally, experiments show that a precise camera calibration is not needed. This is due to the fact that the homography estimation is iteratively performed in conjunction with a closed-loop control law. The main hypothesis of our actual scheme is that the image points used to estimate the homography correspond to coplanar 3D points. Future work will thus concern the generalization of homography estimation in the case of any non-planar target using algorithms such as those presented in [3] and [1].

Acknowledgements

This work was supported by the national French Company of Electricity Power: EDF. We are grateful to the team manager and the researchers of the Teleoperation/Robotics group, at DER Chatou, for their participation and help.

References

[1] B. Boufama, R. Mohr. Epipole and fundamental matrix estimation using the virtual parallax property. In *IEEE International Conference on Computer Vision*, pages 1030–1036, Cambridge, USA, 1995.

[2] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. *Workshop on Vision and Control*, Block Island, USA, To appear, June 1997.

[3] B. Couapel, K.E. Bainian. Stereo vision with the use of a virtual plane in the space. *Chinese Journal of Electronics*, 4(2):32–39, April 1995.

[4] D. Dementhon, L.. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995.

[5] B. Espiau, F. Chaumette, P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[6] B. Espiau. Effect of camera calibration errors on visual servoing in robotics. In *3rd International Symposium on Experimental Robotics*, Kyoto, Japan, October 1993.

[7] O. Faugeras, F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.

[8] K. Hashimoto, ed. *Visual Servoing: Real Time Control of Robot manipulators based on visual sensory feedback*, volume 7 of *World Scientific Series in Robotics and Automated Systems.* World Scientific Press, Singapore, 1993.

[9] S. Hutchinson, G.D. Hager, P.I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.

[10] W.J. Wilson, C.C. Williams Hulls, G.S. Bell. Relative end-effector control using cartesian position-based visual servoing. *IEEE Trans. on Robotics and Automation*, 12(5):684–696, October 1996.

[11] L.E. Weiss, A.C. Sanderson, C.P. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(5):404–417, October 1987.