Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems

A contour approach for image-based control on objects with complex shape

Christophe Collewet

Cemagref Rennes
17 Avenue de Cucillé
35044 Rennes Cedex, France
christophe.collewet@cemagref.fr

François Chaumette

IRISA / INRIA Rennes
Campus Universitaire de Beaulieu
35042 Rennes Cedex, France
francois.chaumette@irisa.fr

Abstract

We describe a method to achieve robotic positioning tasks by image-based visual servoing when the object being observed has a complex and unknown shape. We first focus on the computation of an analytical expression of the interaction matrix according to a polar description of the image contour of the object. Experimental results are presented to validate the proposed algorithm. In particular, the robustness of the control law is tested with respect to a coarse calibrated system, to an approximation of the depth of the object, and to partial occlusion.

1 Introduction

Image-based visual servoing is now a classical technique in robot control. In such a control scheme a matrix, called interaction matrix (or image jacobian), is required. In cases of points or simple geometric primitives (*e.g.* lines or ellipses) this matrix is analytically known. This is why such primitives are most often used [1, 2, 3]. In [2], a general method to obtain the interaction matrix when the object can be represented by a parametric equation has been proposed. This method has been applied to simple objects (lines, circles, cylinders and spheres). However, it cannot be used when no analytical representation of the 3D considered object is provided, which is the case in most real cases.

Several works have involved visual features able to describe an image overall [4, 5, 6, 7]. However these approaches are all based on a learning stage. New learning phases are therefore necessary when the robot has to deal with a new object. Such phases may be heavy [5] even when the nature of the visual features is unchanged. In fact, the way such features change cannot be expressed analytically when the robot moves. Moreover, it is impossible to provide proof of the stability of the system. It would be valuable, from theoretical and practical points of view, to develop control laws based on an analytic approach. This is why a contour approach seems to be interesting. Such an approach has been proposed in [8, 9]. The visual features used in the control scheme are the parameters of an affine deformation of the contour from the current to the desired images. Unfortunately, this approach leads to a singularity when the object and the image plane are parallel. Furthermore, the matching between the two contours may be complex. We will see that these problems do not occur in the method proposed here.

In this paper, we first recall some fundamentals of image-based visual servoing in Section 2. A new approach to interaction matrix computation, based on a polar description of the image of the object, is presented in Section 3. In Section 4, we describe how to extract the visual features from the image. Conditions of convergence are pointed out in Section 5. Finally, the experimental results presented in Section 6 validate our approach.

2 Image-based visual servoing

In image-based visual servoing, the control scheme is performed on the basis of visual features extracted directly from the image: from a m-dimensional vector $\underline{s}$ describing the current visual features the goal is to move the robot so that $\underline{s} = \underline{s}^*$ where $\underline{s}^*$ describes the features when the robot is at its desired position. Such an approach is based on the relationship between the camera velocity T_c and the velocity of the visual

features $\underline{\dot{s}}$. This relationship is described by a matrix called the interaction matrix (or image jacobian):

$$\underline{\dot{s}} = L_{\underline{s}}^T T_c \tag{1}$$

where $T_c = (\underline{V}^T, \underline{\Omega}^T)^T$ with $\underline{V} = (v_x, v_y, v_z)^T$ and $\underline{\Omega} = (\omega_x, \omega_y, \omega_z)^T$ are the translational and rotational components of T_c respectively.

A vision-based task $\underline{e}$ can then be defined by:

$$\underline{e}(\underline{r}, t) = C\,(\underline{s}(\underline{r}, t) - \underline{s}^*) \tag{2}$$

where C is a combination matrix taking into account cases where m is different from n (with n the number of robot d.o.f.). It has to be chosen so that $CL_{\underline{s}}^T$ is full rank and, if $L_{\underline{s}}^T$ is full rank, it can be defined as:

$$C = \widehat{L_{\underline{s}}^{T+}} \tag{3}$$

where $\widehat{L_{\underline{s}}^T}$ is a model or an approximation of $L_{\underline{s}}^T$.

In the case of a motionless object, the camera velocity can be obtained by:

$$T_c = -\lambda \underline{e} \tag{4}$$

This relationship ensures an exponential decrease of $\underline{e}$ ($\underline{\dot{e}} = -\lambda \underline{e}$) when $\widehat{L_{\underline{s}}^T} = L_{\underline{s}}^T$. Otherwise it occurs when the condition $\widehat{L_{\underline{s}}^T}^+ L_{\underline{s}}^T > 0$ is respected [2].

3 Computation of the interaction matrix

Knowing the interaction matrix is essential in image-based visual servoing as explained above. We will see in this section how to compute it in the case of a polar description of the image contour of the object.

Let us consider a point $m = (x, y, z)^T$ expressed in the camera coordinate system (with z axis $\equiv$ optical axis). It projects on the image plane in $M = (X, Y)^T$ that can be expressed in a polar description by:

$$\underline{OM} = (X_c + \rho\cos\theta)\underline{u}_x + (Y_c + \rho\sin\theta)\underline{u}_y \tag{5}$$

where (X_c, Y_c) are the coordinates of the centroid in the image (see Figure 1).

Therefore, the velocity of M can be expressed as:

$$\begin{cases} \dot{X} &= \dot{\rho}\cos\theta - \rho\dot{\theta}\sin\theta + \dot{X}_c \\ \dot{Y} &= \dot{\rho}\sin\theta + \rho\dot{\theta}\cos\theta + \dot{Y}_c \end{cases} \tag{6}$$

As the radius ρ is periodic, we can perform a Fourier expansion:

$$\rho(\theta) = a_0 + \sum_{k=1}^{k=h} a_k \cos k\theta + b_k \sin k\theta \tag{7}$$

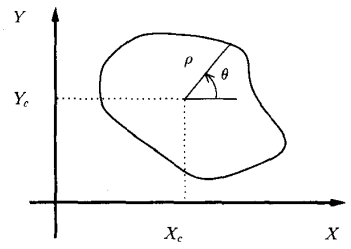

Figure 1: Polar description of a contour

where h is the number of harmonics taken into account.

By denoting $\underline{P} = (a_0, \cdots, a_h, b_1, \cdots, b_h)^T$ and eliminating $\dot{\theta}$, (6) becomes:

$$\dot{X} - \alpha\dot{Y} = \dot{X}_c - \alpha\dot{Y}_c + \beta\frac{\partial\rho}{\partial\underline{P}}\underline{\dot{P}} \tag{8}$$

where we have:

$$\alpha = \frac{\dfrac{\partial\rho}{\partial\theta}\cos\theta - \rho\sin\theta}{\dfrac{\partial\rho}{\partial\theta}\sin\theta + \rho\cos\theta} \text{ and } \beta = \cos\theta - \alpha\sin\theta$$

We then select the vector $\underline{Q} = (X_c, Y_c, \underline{P}^T)^T$ for the visual features. Expression (8) can therefore be expressed as:

$$\dot{X} - \alpha\dot{Y} = \left(1, -\alpha, \beta\frac{\partial\rho}{\partial\underline{P}}\right)\underline{\dot{Q}} \tag{9}$$

If we choose $m = 2h + 3$ independent pairs (ρ, θ) and inverse the system obtained, we have:

$$\underline{\dot{Q}} = M \begin{pmatrix} \dot{X}_1 - \alpha_1\dot{Y}_1 \\ \vdots \\ \dot{X}_m - \alpha_m\dot{Y}_m \end{pmatrix} \tag{10}$$

in which:

$$M = \begin{pmatrix} 1 & -\alpha_1 & \beta_1 \left.\dfrac{\partial\rho}{\partial\underline{P}}\right|_{\rho_1,\theta_1} \\ \vdots & \vdots & \vdots \\ 1 & -\alpha_m & \beta_m \left.\dfrac{\partial\rho}{\partial\underline{P}}\right|_{\rho_m,\theta_m} \end{pmatrix}^{-1} \tag{11}$$

Thereafter, by substituting $\dot{X}$ and $\dot{Y}$ where $\dot{X} = L_X^T T_c$ and $\dot{Y} = L_Y^T T_c$ according to the well known

equations:

$$\begin{pmatrix} L_X^T \\ L_Y^T \end{pmatrix} = \begin{pmatrix} -\frac{1}{z} & 0 & \frac{X}{z} & XY & -1-X^2 & Y \\ 0 & -\frac{1}{z} & \frac{Y}{z} & 1+Y^2 & -XY & -X \end{pmatrix} \tag{12}$$

we obtain the expected expression:

$$L_{\underline{Q}}^T = M \begin{pmatrix} L_{X_1}^T - \alpha_1 L_{Y_1}^T \\ \vdots \\ L_{X_m}^T - \alpha_m L_{Y_m}^T \end{pmatrix} \tag{13}$$

After the manipulations detailed in [10], we obtain the form given in (14) (see the next page) where we have assumed $z_j = \hat{z}^*$, for any $j = 1$ to m, with $\hat{z}^*$ an approximation of the object depth at the desired location. In this expression, we have: $\varphi_i = \varphi_i(X_c, Y_c, \underline{\Theta}, \underline{M}_i)$, $\psi_i = \psi_i(X_c, Y_c, \underline{\Theta}, \underline{M}_i)$, $\underline{\Theta} = (\theta_1, \ldots, \theta_m, \rho_1, \ldots, \rho_m)^T$, $\underline{M}_i = (M_{i,1}, \ldots, M_{i,m})^T$ the expressions of which are complicated and not useful to detail.

From expression (14), we can derive two important points:

- The first and second rows of $L_{\underline{Q}}^T\Big|_{\hat{z}^*}$ show that the interaction matrix associated with the visual feature (X_c, Y_c) differs from that associated with the projection of a point only by adding the terms φ_1, ψ_1, φ_2 and ψ_2, *i.e.* on terms related to the rotation around x and y axes.
- As anticipated, the visual features described by $\underline{P}$ do not vary during x or y translations of the robot. The shape of the contour is thus unchanged during such motions. Moreover, both translations are decoupled.

Once we have obtained the interaction matrix, we shall now examine how to extract the visual features in practice.

4 Extracting the visual features

The problem is to obtain the values of the vector components given by:

$$\underline{s} = (X_c, Y_c, a_0, \cdots, a_h, b_1, \cdots, b_h)^T \tag{15}$$

In order to simplify the problem of image analysis and to focus only on the control law, we used binary objects such as the object represented in Figure 2. In this case, computing (X_c, Y_c) is simple and we therefore only discuss the way to proceed for the a_k's and the b_k's. Theoretically, $2h+1$ points extracted from the contour are sufficient. However, these points are necessarily noised; moreover, we can assume that the number N of points available is such that $N \gg 2h+1$ and a least squares method has therefore been used.

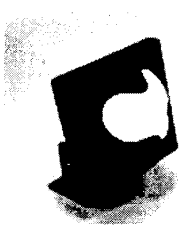

Figure 2: The non-planar object used.

5 Conditions of convergence with respect to the number of harmonics

Concerning the behaviour of the control law with regards to the number of harmonics, the first constraint for h is to achieve the positioning task. We have therefore to ensure that $\widehat{L^T}_{|\underline{s}=\underline{s}^*}$ is full rank 6. It leads to the necessary condition $2h+3 \geq 6$. The smaller value is thus theoretically $h = 2$. However, we will see in the next section that a higher value of h has to be chosen.

We also have to note that the visual features $\underline{s}$ appear in the algorithm in two places:

- in the control law: it must satisfy:

$$\underline{s}|_{k+1} = \underline{s}|_k + \Delta t L_{\underline{s}}^T T_c \tag{16}$$

  where Δt is the control scheme period.

- in the modelling of the image contour: the value of the visual features is computed to minimize the distance between the contour extracted from the image and the modellized contour.

These two points can be summarized by:

$$\underline{s}|_{k+1} = \underline{s}|_k + \Delta t L_{\underline{s}}^T T_c + \delta \underline{s}|_{k+1} \tag{17}$$

where $\delta \underline{s}|_{k+1}$ is a term introduced to describe how the modelling between k and $k+1$ changes (if $h \to \infty$, we have $\delta \underline{s}|_{k+1} \to 0$). This is why h has to be as higher as possible. Otherwise, local minima may occur. In such cases, the last and desired images are very similar but a positioning error exists. We could impose a very high

$$\widehat{L_{\underline{Q}}^T}\Big|_{\hat{z}^*} = \begin{pmatrix} -1/\hat{z}^* & 0 & X_c/\hat{z}^* & X_cY_c+\varphi_1 & -1-X_c^2+\psi_1 & Y_c \\ 0 & -1/\hat{z}^* & Y_c/\hat{z}^* & 1+Y_c^2+\varphi_2 & -X_cY_c+\psi_2 & -X_c \\ 0 & 0 & a_0/\hat{z}^* & \varphi_3 & \psi_3 & 0 \\ 0 & 0 & a_1/\hat{z}^* & \varphi_4 & \psi_4 & b_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a_k/\hat{z}^* & \varphi_{k+3} & \psi_{k+3} & kb_k \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a_h/\hat{z}^* & \varphi_{h+3} & \psi_{h+3} & hb_h \\ 0 & 0 & b_1/\hat{z}^* & \varphi_{h+4} & \psi_{h+4} & -a_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & b_k/\hat{z}^* & \varphi_{k+h+3} & \psi_{k+h+3} & -ka_k \\ 0 & 0 & b_h/\hat{z}^* & \varphi_{2h+3} & \psi_{2h+3} & -ha_h \end{pmatrix} \quad (14)$$

value for h to ensure low enough $\delta\underline{s}|_{k+1}$, but this would require considerable computational time incompatible with running at a video rate.

Typically, as will be shown in the experimental results, a value of h around 25 gives satisfactory results.

6 Experimental results

Here we show the behaviour of the following classical control law:

$$T_c = -\lambda \left(L_{\underline{s}}^T\Big|_{z=\hat{z}^*, \underline{s}=\underline{s}^*} \right)^+ (\underline{s} - \underline{s}^*) \quad (18)$$

Only the most significant results are reported here. More details and other experiments are described in [10].

The first experiment consists of achieving a positioning task using a planar object of complex shape. Figure 3a shows the first image and the desired position of the object, Figure 3b the last image. Figures 3c and 3d show the components of T_c and the normalized error (defined by $\frac{\|\underline{s}-\underline{s}^*\|_k}{\|\underline{s}-\underline{s}^*\|_0}$) respectively. For this experiment h was fixed to 20. This value yielded Δt = 160 ms on a Pentium at 200 Mhz. The desired pose was 15° above the object and the displacement to reach the desired position was around (15°, 15°, 5°, 25 cm, 10 cm, 6 cm) according to the x, y and z axes of the camera. The positioning error was lower than 0.5° for rotations, 1 mm for translations. These results show that the desired location was precisely reached without any particular difficulty. In particular, the assumption z is constant in (18) has therefore no importance.

The second experiment consists of moving the robot in front of a non-planar object. The maximum difference in depth between contour points was nearly 4 cm (see Figure 2). In this case, satisfactory results were obtained with h = 25 (Δt = 240 ms) but not for h = 20 (h = 25 was also suitable for the planar object). The displacement was around (20°, 20°, 20°, 36 cm, 17 cm, 11 cm). The results are depicted in Figure 4. Again, although we supposed in (18) that the object was planar, the desired location was reached. Similar positioning errors have been obtained as previously). Our approach is thus robust to such an assumption.

We next introduced errors in the camera model. Instead of using the intrinsic parameters obtained from a calibration technique, we used the parameters provided by the constructor and assumed that the principal point was in the center of the image. Moreover, we neglected the radial lens distortion. We also introduced errors in the transformation matrix between the robot and the camera (5 degrees for rotations, 3 cm for translations). Experimental results showed that these errors had no significant effects on the positioning error [10]. However, the decrease of the normalized error is faster with a calibrated system (see Figure 5).

We then examined the influence of $\hat{z}^*$ on the control law. A wide range of $\hat{z}^*$ (from 30 cm to 250 cm while the real value was approximately 65 cm) made it possible to reach the desired position. However $\hat{z}^*$ had an effect on the trajectory of the robot (see Figure 6). A high value led to a too long path. Moreover, as we can see on Figure 7, a high value led to high magnitudes of the command while a low value led to low magnitudes (to be compared with Figure 4 where $\hat{z}^*$ = 65 cm). This phenomenon can easily be explained by observing the terms in $1/\hat{z}^*$ in (14) which become terms in $\hat{z}^*$ in the pseudo-inverse needed for (18).

Another experiment consists of realizing a partial occlusion at the beginning of the motion (see Figure 8).

It showed that such an occlusion was not critical: the command was disturbed but, when the occlusion disappeared, the convergence was ensured again.

7 Conclusion and future works

In this paper, we focused on a method to achieve positioning tasks with respect to objects of complex shape. We analytically computed the interaction matrix related to a polar description of the object contour. The experimental results validated the proposed approach. They showed that convergence can be obtained under very low hypotheses (but still within the usual cases of convergence of 2D visual servoing, *i.e.* when $\underline{s} - \underline{s}^*$ is not too large). Only current and desired images are necessary, and no precise knowledge about the object shape and range is required. Moreover, experimental results have shown the robustness of the control law with respect to approximations of the depths, certain partial occlusions and a coarsely calibrated system.

If we compare our approach to an approach based on points of interest, since we can use many contour points, noise has low effect on our control law. On the other hand, a further problem might occur, *i.e.* the matching of points between images. This problem is complicated, especially between very different images such as the first and the desired image. This problem does not exist with our contour-based approach.

However, we can regret heavy computational cost of our method when h is high with consequences on the dynamic performances of the robot (typically, we recall that $\Delta t = 240$ ms for $h = 25$ and a binary object). In the future, a suitable approach to improve the computational cost of image processing might involve using parametric active contours such as those described in [11]. It should be possible to compute our visual features based on a polar description without any particular difficulty. In addition, such an approach should be more robust to extract contours in a significantly noised image of a real object.

References

[1] J. T. Feddema, C. S. G. Lee, and O. R. Mitchell, "Automatic selection of image features for visual servoing of a robot manipulator," in *IEEE Int. Conf. on Robotics and Automation, ICRA'89*, Washington, May 1989, pp. 832–837, USA.

[2] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, June 1992.

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

[4] Z. Bien, W. Jang, and J. Park, "Characterisation and use of feature-jacobian matrix for visual servoing," in *Visual Servoing*, K. Hashimoto, Ed., vol. 7, pp. 317–363. World Scientific, Singapore, 1993.

[5] K. Deguchi and T. Noguchi, "Visual servoing using eigenspace method and dynamic calculation of interaction matrices," in *Int. Conf. on Pattern Recognition, ICPR'96*, 1996, pp. 302–306.

[6] G. Wells, C. Venaille, and C. Torras, "Promising research vision-based robot positioning using neural networks," *Image and Vision Computing*, vol. 14, pp. 715–732, 1996.

[7] S. K. Nayar, S. A. Nene, and H. Murase, "Subspace methods for robot vision," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 750–758, October 1996.

[8] C. Colombo and B. Allotta, "Image-based robot task planning and control using a compact visual representation," *IEEE Trans. on Systems, Man, and Cybernetics — Part A: Systems and Humans*, vol. 29, no. 1, pp. 92–100, January 1999.

[9] T.W. Drummond and R. Cipolla, "Visual tracking and control using lie algebras," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'99*, Fort Collins, Colorado, June 1999, vol. II, pp. 652–657, USA.

[10] C. Collewet, *Contributions à l'élargissement du champ applicatif des asservissements visuels 2D*, Ph.D. thesis, Rennes I University, France, February 1999.

[11] B. Bascle and R. Deriche, "Energy-based methods for 2d curve tracking, reconstruction and refinement of 3d curves and applications," in *SPIE Proceedings, Geometric Methods in Computer Vision*, San Diego, California, July 12-13 1993, vol. 2031, USA.

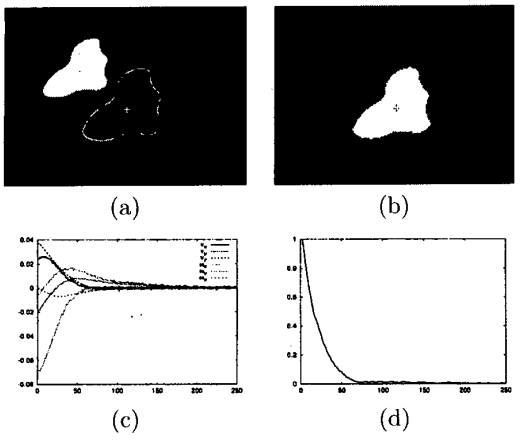

Figure 3: Positioning task with respect to a planar object: (a) First and desired images, (b) Last image, (c) Components of T_c (m/s and rad/s), (d) Normalized error.

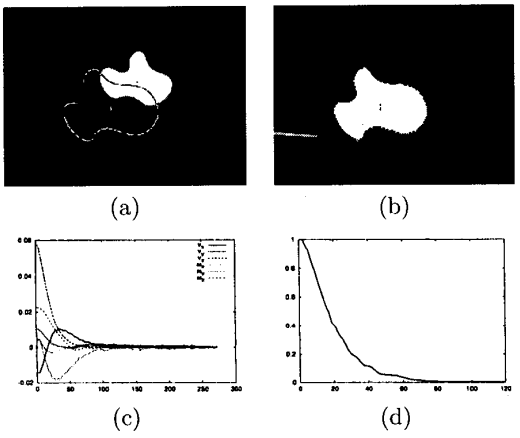

Figure 4: Positioning task with respect to a non-planar object

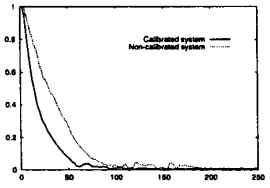

Figure 5: Influence of calibration errors

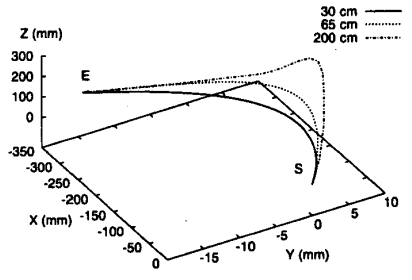

Figure 6: Influence of $\hat{z}^*$ on the robot trajectory (S: start, E: end)

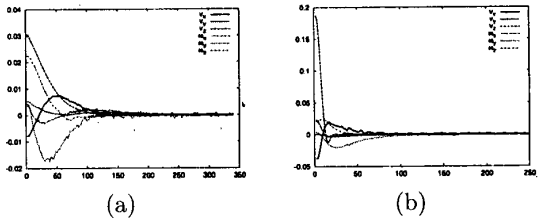

Figure 7: Influence of $\hat{z}^*$ on T_c: (a) $\hat{z}^* = 30$ cm, (b) $\hat{z}^* = 200$ cm

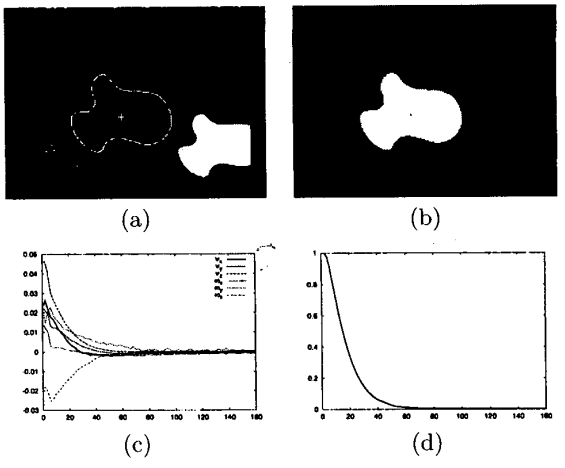

Figure 8: Occlusion test on a non-planar object