Image-based visual servoing on planar objects of unknown shape
Christophe Collewet, François Chaumette, P. Loisel

To cite this version:
Christophe Collewet, François Chaumette, P. Loisel. Image-based visual servoing on planar objects of unknown shape. IEEE Int. Conf. on Robotics and Automation, ICRA’01, 2001, Seoul, South Korea. pp.247-252. inria-00352119

HAL Id: inria-00352119
https://inria.hal.science/inria-00352119
Submitted on 12 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Image-based visual servoing on planar objects of unknown shape

Christophe Collewet  François Chaumette  Philippe Loisel

Cemagref  IRISA / INRIA Rennes  Cemagref
17 Avenue de Cucillé  Campus Universitaire de Beaulieu  17 Avenue de Cucillé
35044 Rennes Cedex, France  35042 Rennes Cedex, France  35044 Rennes Cedex, France
christophe.collewet@cemagref.fr  francois.chaumette@irisa.fr  philippe.loisel@cemagref.fr

Abstract

This paper proposes a way to achieve positioning tasks by visual servoing, for any orientation of the camera, when the desired image of the observed object cannot be precisely described. The object is assumed to be planar and motionless but no knowledge about its shape or pose is required. To simplify the problem, first, we treat the case of a threadlike object and then we show how our approach can be generalized to an object with three particular points. The control law is based on the use of 2d visual servoing and on an estimation of a 3d parameter. Experimental results relative to objects of unknown shape are given to validate the approach. In addition, an algorithm to estimate the depth between the object and the camera is provided which leads to the dimensions of the object.

1 Introduction

Visual servoing is now a classical technique in robot control (see [1] for a description of the different approaches). Nevertheless, in the most often case of an "eye-in-hand" system [2, 3, 4], we still cannot achieve positioning tasks with regard to deformable or not well known objects. Such a case appears when we have to treat applications for example in surgical domain, agri-food industry, agriculture or in unknown environments (underwater, space). Indeed, except manufactured goods for which a model often exists, we rarely have a precise description of the object or of the desired visual features.

In the case of a 3d visual servoing two approaches exist. The first one and the most often used, is based on the computation of the pose object/camera and then requires a model of the object. Therefore, this approach cannot be used in our case. The second one is based on 3d reconstruction by dynamic vision [5, 6] but these techniques are currently not accurate enough with regard to the errors of reconstruction. On the other hand, active vision [7, 8, 9, 10, 11] can limit such errors. However, this approach has only been used on very simple objects. Moreover, let us point out that those 3d reconstruction techniques are sensitive to the calibration of the system.

On the other hand, lots of work in 2d visual servoing have shown that the closed-loop system is little sensitive to calibration problems [4, 12, 13]. Similar results have been obtained concerning the recent 2 1/2d approach [14]. However, the 2d approach as well as the 2 1/2d cannot cope with the objects being studied. Indeed, let us consider an accurate positioning task related to such objects, a raw ham for example. Even though these objects are part of a same class, they are different enough not to use unique desired visual features for all the objects in the class. In such applications the desired visual features have to be considered as unknown. Few authors relate such cases. In [15], thanks to dynamic visual features a positioning task consisting in moving the camera in front of a planar object of unknown shape can be achieved. However, such an approach needs particular motion parameters estimation leading currently to high computation duration and, consequently, to a low control scheme rate. Moreover, this approach does not well suit for positioning tasks since a motion is necessary.

The method described in our paper is based on the use of points as visual features, it can achieve positioning tasks when any orientation is required and for the same objects as those used in [15]: planar objects of unknown shape. Our approach needs no assumption about the shape of the observed object but we have to assume the object motionless. The approach is based on the use of 2d visual servoing and on an estimation of a 3d parameter that will be precised below. We will see that this structure maintains the object of interest in the field of view of the camera, does not need a calibrated camera although it provides dimensions of the object and an estimation of the depth. Unfortunately,
it is sensitive to the calibration of the robot. Finally, our method combines in part the advantages of both 2d visual servoing and 3d reconstruction.

This paper is organized as follows: in Section 2, we deal with a threadlike object to achieve the particular positioning task consisting in moving the robot in front of this object. Next, we show in Section 3 how this approach can be readily generalized to an object with three particular points and for any orientation of the camera. Finally, experimental results are given to validate the algorithm.

2 Case of a simple object: the segment

2.1 Task specification

In this section, we propose to move a robot in front of a threadlike object by visual servoing. In order to achieve applications as those described in the previous section, we suppose the length of the object unknown as well as its pose! The object is described by a segment \([m_1, m_2]\) with \(O m_1 = (x_1, y_1, z_1)^T\) and \(O m_2 = (x_2, y_2, z_2)^T\), expressed in the camera frame which is centered in the optical center \(O\). The optical axis of the camera is the \(z\) axis of the camera frame (see Figure 1). This object projects on the image plane by a perspective projection as a segment \([m_1, m_2]\) with \(O m_1 = (x_1, y_1, f)^T\) according to:

\[
OM = \frac{f}{z} Om \tag{1}
\]

where \(f\) is the focal length assumed to be equal to 1.

![Figure 1: Projection of the object on the image plane](image.png)

The task is achieved, i.e. the camera is in front of the object, when the optical axis is orthogonal to \([m_1, m_2]\) and the image centered with respect to the \(y\) axis. In this case, we have \(z_1 = z_2 = z^*\) and \(Y_1 = -Y_2\).

Even though the object is very simple, the task is not obvious to achieve since an infinity of object poses leads to a centered image without achieving the task. Nevertheless, a way to achieve it is to center the image and then, to scan other positions. This is a typical case where the task function approach described in [2] has to be used. So, we will consider as the main task the one which maintains centered the image, and the secondary task the one which scans other positions while ensuring the image centered.

2.2 Control law

2.2.1 Main task

A way to center the image is to choose as visual features the following vector:

\[
\mathbf{s} = (X_1, X_2, Y_1 + Y_2)^T \tag{2}
\]

and as the desired visual features the vector:

\[
\mathbf{s}^* = (0, 0, 0)^T \tag{3}
\]

We immediately obtain the interaction matrix at the desired position \([1, 2]\):

\[
L_{s*} = \begin{pmatrix}
-1/z^* & 0 & 0 & 0 & -1 & Y_1 \\
-1/z^* & 0 & 0 & 0 & -1 & -Y_1 \\
0 & -2/z^* & 0 & 2(1 + Y_1^2) & 0 & 0
\end{pmatrix} \tag{4}
\]

where \(z^*\) is an approximation of \(z^*\).

2.2.2 Secondary task

At this step, we want to equate \(z_1\) to \(z_2\) while ensuring the image centered. Therefore, such a motion has to belong to \(\text{Ker } L_{s*}^T\) to ensure \(\mathbf{s} = \mathbf{s}^*\). Thereafter, we can apply a control law of the following form to achieve the task:

\[
T_c = -\lambda L_{s*}^T (\mathbf{s} - \mathbf{s}^*) + T_s \tag{5}
\]

with:

\[
T_s = \left(0 \quad \tilde{z}^* \omega_x (1 + Y_1^2) \quad 0 \quad \omega_x \quad 0 \quad 0\right)^T \tag{6}
\]

which belongs to \(\text{Ker } L_{s*}^T\) and in which \(\omega_x\) has to be determined.

If \(\mathbf{s}\) is in the neighborhood of \(\mathbf{s}^*\), only \(T_s\) generates a motion. Intuitively, it is easy to see that the length of the segment \(l = Y_1 - Y_2\) may supply useful information. In particular, we think it takes a maximum when the camera is in front of the object. In order to verify this assumption and to obtain all the parameters needed for (5), we want to know how \(l\) changes during the motion of the camera. Besides, we think that the trajectory of the camera can supply interesting information.
2.3 Modeling

2.3.1 Modeling of the length of the segment

Let us consider the points \( m_1 = (y^*_1, z^*_1)^T \), \( m_2 = (-y^*_2, z^*_2)^T \) and \( m_0 = (0, z_0)^T \) expressed in the camera frame when the camera is at the desired position. We consider that the motion during the maximization of \( l \) results from a \( x \) axis rotation of angle \( \theta_x \), centered in \( m_0 \) (see Figure 2). Therefore, \( m_1 \) and \( m_2 \) can be expressed as: 
\[ m_1' = R (m_1 - m_0) + T \] 
\[ m_2' = R (m_2 - m_0) + T \] 
which yields the projection 
\[ M_1 = (X_1, Y_1)^T \] 
\[ M_2 = (X_2, Y_2)^T \] 
of \( m_1' \) and \( m_2' \) respectively. Then, we search the solution of \( Y_1 = -Y_2 \) with respect to \( z_0 \) to perform a motion which belongs to \( \text{Ker } L'_{\text{image}} \), as seen in 2.2.2. This expression is complicated. Nevertheless, when \( v = y^*/z^* \) is low, it can be expressed at 2nd order by:
\[ z_0 = z^* \tag{7} \]
while \( l \) becomes simply at 3rd order:
\[ l_m = 2v \cos \theta_x \tag{8} \]

2.3.2 Modeling of the trajectory of the camera

Thanks to (7), we can derive the trajectory of the camera with respect to the desired camera position (see Figure 2):
\[ \begin{align*}
    z &= z^* (1 - \cos \theta_x) \\
    y &= z^* \sin \theta_x
\end{align*} \tag{9} \]
Thus the trajectory is simply a circle centered in \((0, z^*)\) 
with a radius \( z^* \):
\[ y^2 - (z - z^*)^2 = z^*^2 \tag{10} \]

2.3.3 Application to control

During a motion, the form which is given by the control law (5), we proceed to an on-line learning of \( l \), supposed to be modeled by (8). In practice, the function we need is \( l_m(\Theta_x - \Theta_{xc}) \) with \( \Theta_x - \Theta_{xc} = \theta_x \) (remember that (8) is obtained with respect to the desired position) and thereafter, \( \Theta_{xc} \) is the unknown value where \( l_m \) takes its maximum. \( l_m(\Theta_x - \Theta_{xc}) \) yields a linear expression in \( \cos \Theta_x \) and \( \sin \Theta_x \) that provides \( \Theta_{xc} \) thanks to a least squares algorithm. Moreover, to ensure an exponential decay of the angular error \( \Theta_x - \Theta_{xc} \), we impose:
\[ \omega_x = K_x (\Theta_x - \Theta_{xc}) \tag{11} \]
with \( K_x \) a positive value.

At the beginning of the motion, when the estimation supplies a not accurate enough value of \( \Theta_{xc} \), we fix \( \omega_x \) to a constant value \( \omega_0 \).
be satisfied simultaneously in the case of any triangle. Therefore, we have arbitrarily chosen to impose $X_1 = X_2$ yielding:

$$ g = (X_1 - X_2, X_1 + X_2, V_1 + V_2)^T \quad (12) $$

$$ g^* = (0, 0, 0)^T \quad (13) $$

$$ L_g^T = \begin{pmatrix} 0 & 0 & 2X_1Y_1 \\ -2/\hat{z} & 0 & X_1(Y_1 - Y_3) - 2(1 + X_1^2)Y_1 + Y_3 \\ 0 & -2/\hat{z} & 2(1 + Y_1^2) \\ -2X_1 & 0 & 2Y_1 \end{pmatrix} \quad (14) $$

and the two vectors which belong to $\text{Ker} \ L_g^T$:

$$ T_{st} = (-\hat{z}^* \omega_x X_1 Y_3, \hat{z}^* \omega_y (1 + X_1^2 + Y_1^2), 0, \omega_x, 0, -\omega_y X_1)^T \quad (15) $$

$$ T_{sh} = (-\hat{z}^* \omega_y (1 + X_1^2), 0, 0, -\omega_y)^T \quad (16) $$

We will see that even if (15) is different from (6), satisfactory results are obtained. The control law is thus:

$$ T_e = -\lambda T_g^T (g - g^*) + T_{st} + T_{sh} \quad (17) $$

In the next section we present the behavior of this control law first on a simple object and next on a more complicated object.

3.2 Experimental results

3.2.1 Experiments on a simple object

The first experiment consists in moving the camera in front of an object with 3 feature points. To simplify the image processing, a binary object has been used. Figures 3a (4a), 3b (4b), 3c (4c) and 3d (4d) depict respectively the behavior of the components of $T_e$, the error in the sensor $\|g - g^*\|$, the desired and current angular values and the measured and modeled image length during the maximization of $l (h)$. Figure 5 summarizes the different phases to achieve the task. Figure 3d (4d) confirms the theoretical results about the modeling of $l = f(\theta_e)$ ($h = f(\theta_e)$). Moreover, on Figure 3c (4c) we remark that we quickly obtain the desired angular value which refines during the estimation and finally stabilizes. In addition, thanks to the least squares algorithm, lets us point out that the noise has not a lot of effect on this value. In other respects, we have applied DeMenthon’s method [16] to obtain the poses of the camera before and after servoing with respect to the object (4 points are then required). The initial pose was around $(25^\circ, 19^\circ, 11^\circ)$ according to the $x$, $y$ and $z$ axes of the camera. The positioning error was around $0.5^\circ$. Therefore, these results are satisfactory.

Besides, as seen in 2.3.2 we have access to $\hat{z}^*$.

### Table 1: Results on 20 experiments.

| $x$ | $\bar{m} = 1/N \sum_i x_i$ | $\max |x_i|$ | $\sigma_x$ | $\min |x_i|$ |
|-----|-----------------|-------------|-----------|-----------|
| $\gamma$ | -0.009 | 0.439 | 0.007 | 0.017 |
| $\beta$ | -0.489 | 0.751 | 0.156 | 0.133 |
| $Q(l)$ | 0.576 | 1.103 | 0.136 | 0.246 |
| $Q(h)$ | 0.562 | 1.104 | 0.140 | 0.242 |
| $Q(z)$ | 0.562 | 1.109 | 0.145 | 0.240 |

In addition, errors have been introduced in the intrinsic parameters of the camera: we have added an error of 10% on these parameters and we have neglected the radial distortion of the lens. Experimental results, consisting in achieving positioning tasks, showed that no significant difference exists concerning the positioning error between a calibrated camera and a coarse calibrated one. This result agrees with the remark made in section 2.3.3.

The second experiment consists in positioning the camera not in front of the object but for a particular desired pose. Since we know the angular value $\Theta_{ze}$ ($\Theta_{ye}$) (see 2.3.3) which leads to move the robot in front of the object, by adding specified constant values any orientations can be reached (see Figures 6 and 7). In this case we obtained $(15^\circ, 19.0^\circ)$ instead of $(15^\circ, 20^\circ)$. Again, these results are satisfactory.

3.2.2 Experiment on a complicated object

In this third experiment, we consider the case of a raw ham. This object is moving along a conveyor and the goal is to place the camera in front of it to perform...
an inspection of its cut by a machine vision. To do this, we used the method described in [17] to select and track the 3 feature points we need. It is based on SSD matching and assumes translational frame-to-frame displacements. The experimental results are depicted on Figures 8 and 9. They lead to similar conclusions as the previous experimental results. However, the algorithm of extraction of the visual features is more noisy and a higher positioning error is obtained but, this time, difficult to measure. Again, 3d features can be obtained to characterize the cut of the ham.

4 Conclusion

We have presented in this paper a way to achieve any 2d visual servoing positioning task in the case of a planar and motionless object of complex shape thanks to simple geometric visual features. Experimental results have shown that accurate positionings can be obtained \( \approx 0.5\% \) even if the camera were not calibrated. In addition, thanks to the trajectory of the camera, the depth between the camera and the object can be obtained yielding precise 3d reconstruction \( \approx 0.6 \% \). Thus, our method combines in part the advantages of both 2d and 3d visual servoing in the sense that 3d information can be obtained with a coarse calibrated camera.

Finally, we think that this work contributes to an expansion of the application area of visual servoings in the sense that now, complex objects or objects of unknown shape can be treated even if the desired image is not precisely known.

References

Figure 3: 1st experiment. Maximization of $l$.

Figure 4: 1st experiment. Maximization of $h$.

Figure 5: 1st experiment. (a) Initial Position. (b) Position after "centering". (c) Position after maximization of $l$. (d) Final position.

Figure 6: 2nd experiment. Servoing of $\Theta_x$.

Figure 7: 2nd experiment. Servoing of $\Theta_y$.

Figure 8: 3rd experiment. Maximization of $l$ and $h$.

Figure 9: 3rd experiment. (a) Initial Position. (b) Position after "centering". (c) Position after maximization of $l$. (d) Final position.