

A first step toward visual servoing using image moments

François Chaumette

► **To cite this version:**

François Chaumette. A first step toward visual servoing using image moments. IEEE/RSJ Int Conf on Intelligent Robots and Systems, IROS'02, 2002, Lausanne, Switzerland, Switzerland. 1, pp.378-383, 2002. <inria-00352087>

HAL Id: inria-00352087

<https://hal.inria.fr/inria-00352087>

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.

A first step toward visual servoing using image moments

François Chaumette

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

Abstract

In this paper, we determine the analytical form of the interaction matrix related to any moments that can be computed from binary or segmented images. We then apply this theoretical result to image-based visual servoing by selecting six combinations of moments able to control the six dof of the system. The experimental results we present show that a correct behavior of the system is obtained if we consider either a simple symmetrical object, either a planar object with complex and unknown shape.

1 Introduction

Image moments have been widely used in computer vision for a very long time, especially for pattern recognition applications [6, 9, 8]. It would be interesting to use them in visual servoing since they provide a generic representation of any object that can be segmented in an image. They also provide a more geometric and intuitive meaning than the features that have been proposed in [3] from the contour of an object. Temptatives of using moments in 2D visual servoing have already been presented in the past (without speaking of course of all the works based on several blobs centroids). The problem was that the analytical form of the interaction matrix related to image moments was not available. This matrix is however essential to design a visual servoing control scheme. That is why, in [1], if the idea of using image moments was expressed rigorously, very coarse approximations were performed in practice to control only 4 dof of a robot using the area, the centroid and the main orientation of an object in the image. Similarly, in [10], a neural networks has been developed to numerically estimate this interaction matrix. The first contribution of this paper is that it presents a method to determine the analytical form of the interaction matrix related to any image moments.

An other objective of using moments in image-based visual servoing is to try to determine features that avoid the potential problems that may appear when

redundant image points coordinates are used: local minima, coupled features that leads to unadequate robot trajectories, etc [2]. In fact, we would like to find again the nice properties of 2 1/2 D visual servoing [7], but using visual features extracted only from the current image (to avoid any partial pose estimation at each iteration of the control scheme, which introduces some sensitivity to image noise). A nice step in that way has been recently presented in [4]. More precisely, we search for six independent visual features such that the corresponding interaction matrix has a maximal decoupled structure, without any singularity, and such that its condition number is as low as possible (to improve the robustness and the numerical stability of the system). The second contribution of this paper is concerned with this objective.

2 Interaction matrix of image moments

Let \mathcal{O} be the observed object and $I(t)$ the image acquired by the camera at time t . We denote $\mathcal{D}(t)$ the part of $I(t)$ where the object projects, and $\mathcal{C}(t)$ the contour of $\mathcal{D}(t)$. We do not consider in this paper the intensity level of each pixel, which means that either binary images are acquired, either a spatial segmentation algorithm, providing binary images, is first performed on the acquired images. In that case, the moments m_{ij} of \mathcal{O} in the image are defined by:

$$m_{ij}(t) = \iint_{\mathcal{D}(t)} f(x,y) dx dy \quad (1)$$

where $f(x,y) = x^i y^j$. We are interested in determining an analytical form describing the time variation \dot{m}_{ij} of moment m_{ij} in function of the relative kinematic screw $\mathbf{T} = (\mathbf{V}^T \ \Omega^T)^T$ between the camera and the object, where \mathbf{V} and Ω represent the translational and rotational velocity components respectively. As for classical geometrical features [5], we will obtain a linear link that can be expressed under the form:

$$\dot{m}_{ij} = \mathbf{L}_{m_{ij}} \mathbf{T} \quad (2)$$

where $\mathbf{L}_{m_{ij}}$ is called the interaction matrix related to m_{ij} .

In (1), the only part that is function of time t is $\mathcal{D}(t)$. The time variation of m_{ij} can thus be obtained from the variation of $\mathcal{C}(t)$. More precisely, we have (see Figure 1):

$$\dot{m}_{ij} = \oint_{\mathcal{C}(t)} f(x, y) \vec{\mathbf{x}} \cdot \vec{\mathbf{n}} dl \quad (3)$$

where $\vec{\mathbf{x}}$ is the velocity of contour point $\mathbf{x} = (x \ y)^T$, $\vec{\mathbf{n}}$ is the unitary vector normal to $\mathcal{C}(t)$ at point \mathbf{x} , and dl is an infinitesimal element of contour $\mathcal{C}(t)$.

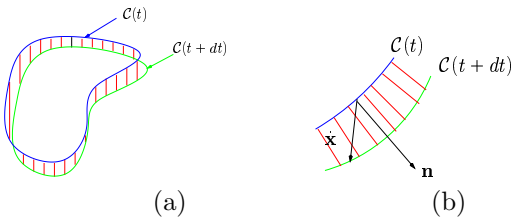


Figure 1: Time variation of contour $\mathcal{C}(t)$

If the following conditions are satisfied (which is the case in practice):

- $\mathcal{C}(t)$ is continuous by parts,
- vector $f(x, y) \vec{\mathbf{x}}$ is tangent to $\mathcal{D}(t)$ and continuously differentiable, $\forall \mathbf{x} \in \mathcal{D}(t)$,

we can use the famous Green's theorem and write (3) under the form:

$$\dot{m}_{ij} = \iint_{\mathcal{D}(t)} \text{div}[f(x, y) \vec{\mathbf{x}}] dx dy \quad (4)$$

By developing (4), we finally obtain:

$$\dot{m}_{ij} = \iint_{\mathcal{D}} \left[\frac{\partial f}{\partial x} \dot{x} + \frac{\partial f}{\partial y} \dot{y} + f(x, y) \left(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} \right) \right] dx dy \quad (5)$$

In this expression, the terms \dot{x} , \dot{y} , $\frac{\partial \dot{x}}{\partial x}$ and $\frac{\partial \dot{y}}{\partial y}$ can be linearly expressed to the kinematic screw \mathbf{T} . Indeed, for any point with coordinates $\mathbf{x} = (x \ y)^T$ in the image whose corresponding 3D point has depth Z , we have the well known relation:

$$\vec{\mathbf{x}} = \mathbf{L}_{\mathbf{x}} \mathbf{T} \quad (6)$$

where:

$$\mathbf{L}_{\mathbf{x}} = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -1-x^2 & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} \quad (7)$$

At this step, we consider that the 3D observed object belongs to a continuous surface, which means in other words that we assume that it does not present any depth discontinuity. In that case, the depth of

any 3D object point can be expressed as a continuous function of its image coordinates x and y :

$$\frac{1}{Z} = \sum_{p \geq 0, q \geq 0} A_{pq} x^p y^q \quad (8)$$

For instance, if the object is planar, or has a planar limb surface (see [5] for more details), whose equation expressed in the camera frame is given by:

$$Z = \alpha X + \beta Y + Z_0,$$

we have:

$$\frac{1}{Z} = Ax + By + C \quad (9)$$

where $A = -\alpha/Z_0$, $B = -\beta/Z_0$ and $C = 1/Z_0$. In the remainder of this paper, we will only consider planar objects. Equation (9) will thus be used instead of (8) to simplify the mathematical developments.

Using (9) in (7), (6) can finally be written:

$$\begin{cases} \dot{x} = -(Ax + By + C).V_x \\ \quad + x(Ax + By + C).V_z \\ \quad + xy.\Omega_x - (1 + x^2).\Omega_y + y.\Omega_z \\ \dot{y} = -(Ax + By + C).V_y \\ \quad + y(Ax + By + C).V_z \\ \quad + (1 + y^2).\Omega_x - xy.\Omega_y - x.\Omega_z \end{cases} \quad (10)$$

from which we deduce:

$$\begin{cases} \frac{\partial \dot{x}}{\partial x} = -AV_x + (2Ax + By + C)V_z + y\Omega_x - 2x\Omega_y \\ \frac{\partial \dot{y}}{\partial y} = -BV_y + (Ax + 2By + C)V_z + 2y\Omega_x - x\Omega_y \end{cases} \quad (11)$$

Substituting (10) and (11) in (5), and knowing that $f(x, y) = x^i y^j$, $\frac{\partial f}{\partial x} = ix^{i-1} y^j$ and $\frac{\partial f}{\partial y} = jx^i y^{j-1}$, we can express (5) under the expected form (2). We obtain after simple developments:

$$\mathbf{L}_{m_{ij}} = (m_{vx} \ m_{vy} \ m_{vz} \ m_{wx} \ m_{wy} \ m_{wz}) \quad (12)$$

where:

$$\begin{cases} m_{vx} = -i(Am_{ij} + Bm_{i-1,j+1} + Cm_{i-1,j}) - Am_{ij} \\ m_{vy} = -j(Am_{i+1,j-1} + Bm_{ij} + Cm_{i,j-1}) - Bm_{ij} \\ m_{vz} = (i+j+3)(Am_{i+1,j} + Bm_{i,j+1} + Cm_{ij}) - Cm_{ij} \\ m_{wx} = (i+j+3)m_{i,j+1} + jm_{i,j-1} \\ m_{wy} = -(i+j+3)m_{i+1,j} - im_{i-1,j} \\ m_{wz} = im_{i-1,j+1} - jm_{i+1,j-1} \end{cases}$$

The time variation of a moment of order $i + j$ can thus be expressed from the moments of order less than $i + j + 2$ and from the 3D parameters A , B and C . This nice property will allow us to easily use image moments in image-based visual servoing.

3 Determination of adequate moments for image-based visual servoing

The main problem is now to determine what are the moments we have to select in a visual servoing control scheme. For that, we first consider the classical moments of low order that are the area $a(=m_{00})$, and the coordinates $x_g(=m_{10}/m_{00})$ and $y_g(=m_{01}/m_{00})$ of the center of gravity of an object in the image.

The interaction matrices related to these visual features are obtained directly from (12):

$$\begin{aligned} \mathbf{L}_a &= \begin{pmatrix} -aA & -aB & a(3/Z_g - C) \\ 3ay_g & -3ax_g & 0 \end{pmatrix} \\ \mathbf{L}_{x_g} &= \begin{pmatrix} -1/Z_g & 0 & x_g/Z_g + \epsilon_1 \\ x_g y_g + n_{11} & -(1+x_g^2+n_{20}) & y_g \end{pmatrix} \\ \mathbf{L}_{y_g} &= \begin{pmatrix} 0 & -1/Z_g & y_g/Z_g + \epsilon_2 \\ 1+y_g^2+n_{02} & -x_g y_g - n_{11} & -x_g \end{pmatrix} \end{aligned} \quad (13)$$

with $1/Z_g = Ax_g + By_g + C$, $\epsilon_1 = An_{20} + Bn_{11}$, $\epsilon_2 = An_{11} + Bn_{02}$ where n_{20}, n_{02} and n_{11} are the normalized centred moments of order 2:

$$n_{ij} = 4\mu_{ij}/a \text{ with } \begin{cases} \mu_{20} = m_{20} - ax_g^2 \\ \mu_{02} = m_{02} - ay_g^2 \\ \mu_{11} = m_{11} - ax_g y_g \end{cases}$$

We can note that \dot{a} depends linearly of a (we have of course $\dot{a} = 0$ if $a = 0$). Furthermore, when the object is centered and parallel to the image plane ($A = B = x_g = y_g = 0$), we have $\dot{a} = 0$ for all camera motions but the expected translation V_z along the optical axis.

We can also note that the interaction matrix related to the coordinates of the center of gravity of the object is a generalization of the interaction matrix (7) related to the coordinates of a point. Indeed, for a physical point, we have $n_{20} = n_{11} = n_{02} = 0$ and we can set $A = B = 0$ in (13) to obtain exactly classical equation (7).

From another point of view, we can say that all previous works in visual servoing that used the coordinates of the center of gravity as visual features ([4] for instance) in fact used with (7) an approximation of the real interaction matrix. However, it was not a crucial problem since closed-loop visual servoing schemes are well known to be robust with respect to such modeling errors.

Finally, it is well known and clear from (13) that x_g and y_g are mainly related to translational motion V_x along camera x -axis and rotational motion ω_y around y -axis (for x_g), and to translational motion V_y and rotational motion ω_x (for y_g).

We now consider the centered moments μ_{ij} . They

are defined by:

$$\mu_{ij} = \iint_{\mathcal{D}} (x - x_g)^i (y - y_g)^j dx dy \quad (14)$$

Knowing the relation between μ_{ij} and the moments m_{kl} of order less than $i + j + 1$:

$$\mu_{ij} = \sum_{k=0}^i \sum_{l=0}^j \binom{i}{k} \binom{j}{l} (-x_g)^{i-k} (-y_g)^{j-l} m_{kl} \quad (15)$$

we can determine the interaction matrix related to μ_{ij} using (12). We can even obtain the analytical form of this matrix expressed with the centered moments by using the reciprocal relation to (15) that links any m_{kl} to the centered moments of order less than $k + l + 1$. After some (tedious) developments, we obtain the following simple form:

$$\mathbf{L}_{\mu_{ij}} = \begin{pmatrix} \mu_{vx} & \mu_{vy} & \mu_{vz} & \mu_{wx} & \mu_{wy} & \mu_{wz} \end{pmatrix} \quad (16)$$

with:

$$\begin{aligned} \mu_{vx} &= -(i+1)A\mu_{ij} - iB\mu_{i-1,j+1} \\ \mu_{vy} &= -jA\mu_{i+1,j-1} - (j+1)B\mu_{ij} \\ \mu_{vz} &= -A\mu_{wy} + B\mu_{wx} + (i+j+2)C\mu_{ij} \\ \mu_{wx} &= (i+j+3)\mu_{i,j+1} + ix_g\mu_{i-1,j+1} \\ &\quad + (i+2j+3)y_g\mu_{ij} - in_{11}\mu_{i-1,j} - jn_{02}\mu_{i,j-1} \\ \mu_{wy} &= -(i+j+3)\mu_{i+1,j} - (2i+j+3)x_g\mu_{ij} \\ &\quad - jy_g\mu_{i+1,j-1} + in_{20}\mu_{i-1,j} + jn_{11}\mu_{i,j-1} \\ \mu_{wz} &= i\mu_{i-1,j+1} - j\mu_{i+1,j-1} \end{aligned}$$

We can first note that all centered moments are invariant with respect to translational motions parallel to the image plane only if the image plane is parallel to the object ($\mu_{vx} = \mu_{vy} = 0$ iff $A = B = 0$).

Second, we could hope that using as visual features \mathbf{s} the six moments of order less than 3 (that is $\mathbf{s} = (a \ x_g \ y_g \ \mu_{20} \ \mu_{11} \ \mu_{02})^T$), the six dof of the system could be controlled. However, we can show that the interaction matrix \mathbf{L}_s related to \mathbf{s} is always of maximal rank 5 when symmetrical objects appears in the image (such as for example circles, ellipses, squares, and rectangles). Furthermore, for non symmetrical objects, even if \mathbf{L}_s is of full rank 6, its condition number is really too high to provide satisfactory results (the control scheme is unstable). Moments of order at least equal to 3 have thus to be involved in the selection of the visual features.

Despite this, an interesting feature can be selected from the moments of order 2. It is of course the object orientation θ defined by:

$$\theta = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \quad (17)$$

We obtain using (16):

$$\mathbf{L}_\theta = \begin{pmatrix} \theta_{vx} & \theta_{vy} & \theta_{vz} & \theta_{wx} & \theta_{wy} & -1 \end{pmatrix} \quad (18)$$

$$\text{with: } \begin{cases} \theta_{vx} &= \alpha A + \beta B \\ \theta_{vy} &= -\gamma A - \alpha B \\ \theta_{vz} &= -A\theta_{wy} + B\theta_{wx} \\ \theta_{wx} &= -\beta x_g + \alpha y_g + \delta \\ \theta_{wy} &= \alpha x_g - \gamma y_g + \nu \end{cases}$$

$$\text{and } \begin{cases} \alpha &= \mu_{11}(\mu_{20} + \mu_{02})/\Delta \\ \beta &= [2\mu_{11}^2 + \mu_{02}(\mu_{02} - \mu_{20})]/\Delta \\ \gamma &= [2\mu_{11}^2 + \mu_{20}(\mu_{20} - \mu_{02})]/\Delta \\ \delta &= 5[\mu_{12}(\mu_{20} - \mu_{02}) + \mu_{11}(\mu_{03} - \mu_{21})]/\Delta \\ \nu &= 5[\mu_{21}(\mu_{02} - \mu_{20}) + \mu_{11}(\mu_{30} - \mu_{12})]/\Delta \\ \Delta &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \end{cases}$$

We can note that θ is invariant with respect to any translational motion when the object plane is parallel to the image plane ($A = B = 0$). Of course, we can also note the direct link between the variation of θ and the rotational motions ω_z around the optical axis.

We thus have four visual features (coordinates x_g and y_g of the center of gravity, area a , and orientation θ) which can be easily obtained from moments of order less than 3 and which are closely related respectively to V_x and ω_y , V_y and ω_x , V_z , and ω_z . These four features have already been used in the partitioned system presented in [4] (with a more intuitive and approximative modeling of the interaction matrix). The question is now to determine two supplementary visual features in order to decouple ω_y from V_x and ω_x from V_y . For that, the classical skewness terms defined by $S_x = \mu_{30}/\mu_{20}^{3/2}$ and $S_y = \mu_{03}/\mu_{02}^{3/2}$ could be used [9, 10]. However, experiments have shown that a larger convergence domain and a better behavior are obtained using features designed from normalized Hu's invariants [6], intuition, and tests. More precisely, we propose for symmetrical objects:

$$\begin{cases} s_x = \sqrt{a} (s_1 t_1 + s_2 t_2)/K \\ s_y = \sqrt{a} (s_2 t_1 - s_1 t_2)/K \end{cases} \quad (19)$$

$$\text{where } \begin{cases} s_1 = \mu_{03} - 3\mu_{21}, t_1 = (\mu_{20} - \mu_{02})^2 - 4\mu_{11}^2 \\ s_2 = \mu_{30} - 3\mu_{12}, t_2 = 4\mu_{11}(\mu_{20} - \mu_{02}), \\ K = \Delta (\mu_{20} + \mu_{02})^{3/2} \end{cases}$$

and, for non symmetrical objects:

$$\begin{cases} P_2 = \Delta/(\mu_{20} + \mu_{02})^2 \\ P_3 = a (s_1^2 + s_2^2)/(\mu_{20} + \mu_{02})^3 \end{cases} \quad (20)$$

4 Experimental results

In the presented experimental results, we have used the classical control law:

$$\mathbf{T}_c = -\lambda \mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+ (\mathbf{s} - \mathbf{s}^*) \quad (21)$$

where \mathbf{T}_c is the camera kinematic screw sent to the low level robot controller, λ is a positive gain, and $\mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+$ is the pseudo-inverse of the interaction matrix related to \mathbf{s} computed for the desired value \mathbf{s}^* . When six visual features are used, we simply have $\mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+ = \mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^{-1}$ since \mathbf{s} is chosen such that $\mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}$ is of full rank 6. Using this control scheme, only the local stability of the system is demonstrated since the sufficient positivity condition:

$$\mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+ \mathbf{L}_{\mathbf{s}(\mathbf{t})} > 0$$

is only ensured in a neighborhood of $\mathbf{s} = \mathbf{s}^*$ (see [5] for more details). Using $\mathbf{L}_{\mathbf{s}}^+$ instead of $\mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+$ in control scheme (21) would may give even better results, but we have currently no way to estimate the 3D parameters A, B and C involved in the interaction matrix when objects of unknown shape are considered.

4.1 Case of a simple symmetrical object

We first consider a simple rectangular object whose corners can easily be tracked in the acquired images. The desired camera position is such that the rectangle is parallel to the image plane, at a range of 50 cm from the camera optical center ($A = B = 0, C = 2$), and such that the rectangle is centered and horizontal in the image (see Figure 2.a). The displacement that the camera has to realize is approximatively composed of a rotation of -35, -20 and -65 degrees around camera axis x, y and z respectively, and of a translation of 10, -25 and 15 cm along these axes. The image acquired at the initial camera position is displayed on Figure 2.b.

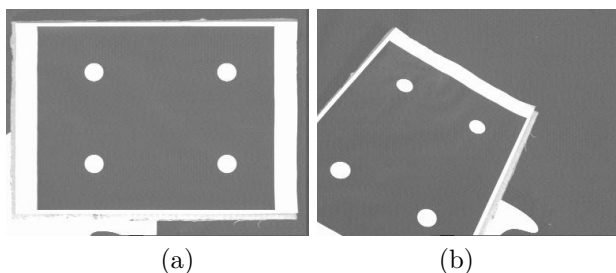


Figure 2: (a) Desired image, (b) Initial image

Using the coordinates x_i and y_i of the four corners as visual features, the related interaction matrix computed for the desired value \mathbf{s}^* does not present any interesting decoupling properties and we can note that its condition number is approximatively equal to 180. The obtained results are depicted on Figure 4. Hopefully, the camera reaches its desired position, but we can see that the corners trajectories in the image, as well as the camera trajectory (that can be extrapolated from the computed velocity), are not particularly satisfactory.

We now describe the results obtained for the same experiment exactly but using the six visual features described in Section 3. In practice, we determine the shape of the quadrilateral that corresponds to the image of the rectangle from the position of the corners, and then compute the moments of the quadrilateral. We thus have as visual features:

$$\mathbf{s} = (x_g \quad y_g \quad a \quad s_x \quad s_y \quad \theta)^T$$

whose desired value \mathbf{s}^* is given by:

$$\mathbf{s} = (0 \quad 0 \quad a^* \quad 0 \quad 0 \quad 0)^T$$

For that value, the interaction matrix presents excellent decoupling properties since it is upper triangular with only two non null values in the upper part. More precisely, we have:

$$\mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*} = \begin{pmatrix} -2 & 0 & 0 & 0 & -1.02 & 0 \\ 0 & -2 & 0 & 1.01 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.19 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

whose condition number is now equal to 17.1. The numerical stability of the system is thus greatly improved using moments-based features instead of points features (gain with scale equal to more than 10). The obtained results are depicted on Figure 5 and 3. By comparing Figure 4 and 5, we can immediately observe the improvements obtained in the camera and image corners trajectories.

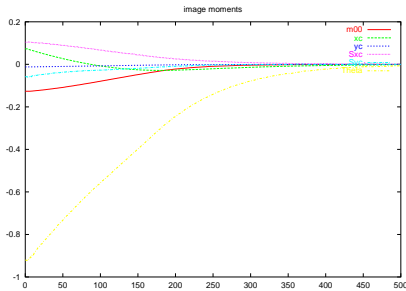


Figure 3: Results for the rectangle using image moments as visual features: Image moments error (m)

4.2 Case of a complex object: the “whale”

We finally present the results obtained for a non symmetrical planar object whose shape is quite complex (see Figure 6.a and 6.b). In practice, the moments are now computed by simple summations on the binarized images, this simple image processing being performed at the video rate. We now use as visual features:

$$\mathbf{s} = (x_g \quad y_g \quad a \quad P_2 \quad P_3 \quad \theta)$$

The numerical value of the interaction matrix computed from the desired image given in Figure 6.a is given by:

$$\mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*} = \begin{pmatrix} -2 & 0 & 0.09 & 0.01 & -1.04 & 0.01 \\ 0 & -2 & 0.01 & 1.02 & -0.01 & -0.05 \\ 0 & 0 & 0.26 & 0.01 & -0.01 & 0 \\ 0 & 0 & 0 & 0.08 & -0.11 & 0 \\ 0 & 0 & 0 & 0.04 & 0.05 & 0 \\ 0 & 0 & 0 & -0.00 & -0.05 & -1 \end{pmatrix}$$

We can note that $\mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}$ is no more purely triangular (but such a form has not been obtained yet) and presents more non null values in the upper part than previously (mainly because the “whale” is not centered in the desired image). The condition number of $\mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}$ is now equal to 42.1, which is still far better than using points coordinates in \mathbf{s} (and it would be impossible to use points coordinates in that case).

The results obtained for the same displacement to realize as before are depicted on Figure 6. Even if the decreasing behavior of the visual features is similar for the rectangle and for the “whale” (compare Figure 3 and 6.c), the camera trajectory is a little bit less satisfactory (compare Figure 5.c and 6.d). The convergence is however obtained, even for the large considered displacement, and all the “whale” remains in the camera field of view during the positioning task. We can finally note that the small white part outside the “whale” (see Figure 6.a and 6.b), which is taken into account in the computation of the moments, does not perturb at all the behavior of the system.

5 Conclusion

This paper presents a first step to use image moments in visual servoing. The improvements with respect to classical visual servoing seem to be significant since we obtained a generic representation with nice decoupling and stability properties. A lot of works remain however to be done. First, we would like to determine other moments-based visual features to obtain even better results, especially for non symmetrical objects. We also have to study the robustness of our scheme with respect to calibration errors and to partial occlusions.

References

- [1] Z. Bien, W. Jang, J. Park: “Characterization and use of feature-jacobian matrix for visual servoing”, in Visual servoing, pages 317-363, World Scientific, 1993.
- [2] F. Chaumette: “Potential problems of stability and convergence in image-based and position-based visual servoing”, in The Confluence of Vision and Control, pages 66-78, LNCIS Series, No 237, Springer-Verlag, 1998.

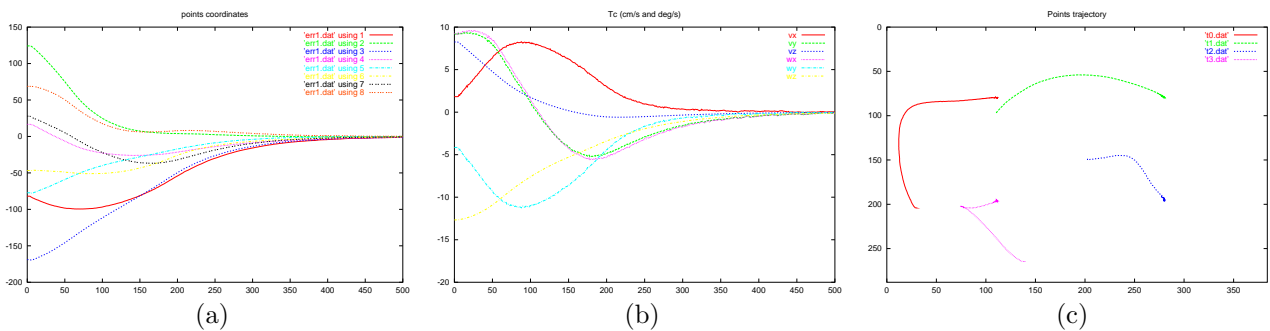


Figure 4: Results for the rectangle using point coordinates as visual features: (a) Point coordinates error (pixels) versus iteration number, (b) Computed camera velocity (cm/s and dg/s) versus iteration number, (c) Image points trajectory

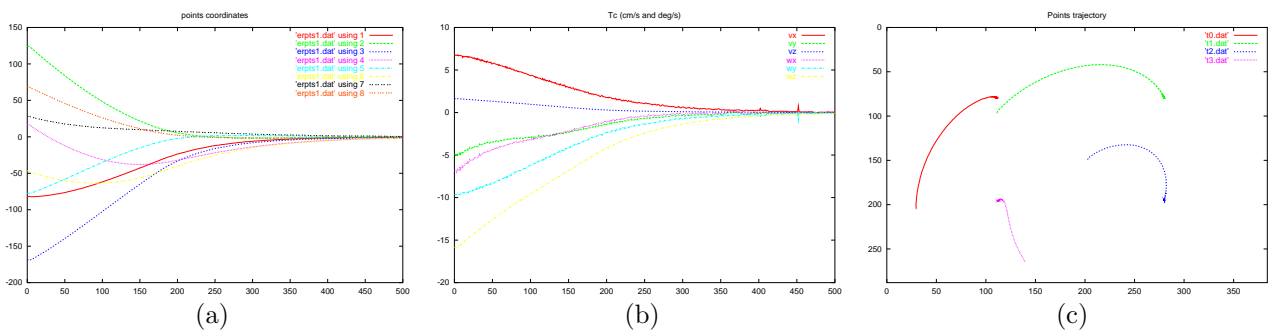


Figure 5: Results for the rectangle using image moments as visual features: (a) Point coordinates error (pixels), (b) Computed camera velocity (cm/s and dg/s), (c) Image points trajectory

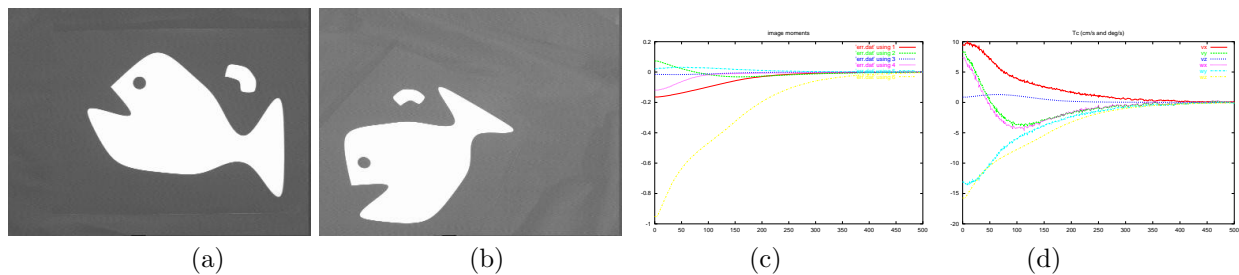


Figure 6: Results for the “whale”: (a) Desired image, (b) Initial image, (c) Image moments error (m), (d) Computed camera velocity (cm/s and dg/s)

[3] C. Collewet, F. Chaumette: “A contour approach for image-based control of objects with complex shape”, IROS’2000, Volume 1, pages 751-756, Takamatsu, Japan, November 2000.

[4] P. Corke, S. Hutchinson: “A new partitioned approach to image-based visual servo control”, IEEE Trans. on Robotics and Automation, 17(4):507-515, August 2001.

[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] M.K. Hu: “Visual pattern recognition by moment invariants”, IRE Trans. on Information Theory, 8:179-187, 1962.

[7] E. Malis, F. Chaumette, S. Boudet: “2 1/2 D Visual Servoing”, IEEE Trans. on Robotics and Automation, 15(2):238-250, April 1999.

[8] R. Mukundan, K.R. Ramakrishnan: “Moment functions in image analysis: theory and application”, World Scientific, 1998.

[9] R.J. Prokop, A.P. Reeves: “A survey of moment-based techniques for unoccluded object representation and recognition”, CVGIP, 54(5):438-460, September 1992.

[10] G. Wells, C. Venaille, C. Torras: “Vision-based robot positioning using neural networks”, Image and Vision Computing, 14:715-732, December 1996.