

Asservissement visuel à partir de moments invariants

Application of Moment Invariants to Visual Servoing

O. Tahri

F. Chaumette

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

E-mail: Prenom.Nom@irisa.fr

Résumé

Dans cet article, on présente une méthode basée sur les moments invariants pour minimiser les non linéarités de la matrice d'interaction des primitives choisies et pour obtenir un asservissement visuel 2D découplé. Des résultats expérimentaux utilisant une caméra embarquée sur un robot à 6 ddl pour se positionner parallèlement à un objet plan de forme complexe sont présentés pour démontrer l'efficacité de la méthode proposée.

Mots Clés

Moments invariants, commande découplée, non linéarité

Abstract

In this paper, we present how moment invariants can be used to design an uncoupled 2D visual servoing scheme and to minimize the nonlinearity of the interaction matrix related to the selected visual features. Experimental results using a 6 dof eye-in-hand system to position a camera parallel to planar objects of complex shape are presented to demonstrate the efficiency of the proposed method.

Keywords

Moment invariants, decoupled control, nonlinearity.

1 Introduction

En asservissement visuel 2D [8, 10], la commande des mouvements du robot est réalisée en utilisant directement des informations visuelles extraites dans l'image acquise par une (ou plusieurs) caméra (s). Soit \mathbf{s} un ensemble de k primitives qui caractérisent l'objet dans l'image. La vitesse de ces primitives et les vitesses appliquées sur la caméra sont liées par la formule classique :

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{T} \quad (1)$$

où \mathbf{T} est le vecteur de vitesses relatives entre la caméra et l'objet. La matrice \mathbf{L}_s est appelée matrice jacobienne ou matrice d'interaction associée à \mathbf{s} . Le schéma de commande consiste à annuler une fonction de tâche :

$$\mathbf{e} = \mathbf{C}(\mathbf{s} - \mathbf{s}^*) \quad (2)$$

où \mathbf{s} est l'état courant du vecteur de primitives et \mathbf{s}^* son état désiré. Pour commander les 6 degrés de liberté de la

caméra on utilise en général $\mathbf{C} = \mathbf{I}_6$ si $k = 6$. Si on désire obtenir une décroissance exponentielle de la fonction de tâche :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \quad (3)$$

où λ est un gain scalaire, la loi de commande à utiliser pour une caméra embarquée observant un objet statique est donnée par :

$$\mathbf{T}_c = -\lambda \widehat{\mathbf{L}}_s^+ \mathbf{e} \quad (4)$$

où $\mathbf{T}_c = (V_X, V_Y, V_Z, \Omega_X, \Omega_Y, \Omega_Z)^T$ est le vecteur vitesse à appliquer à la caméra, $\widehat{\mathbf{L}}_s$ est une estimation de la matrice \mathbf{L}_s , et $\widehat{\mathbf{L}}_s^+$ est la pseudo inverse de la matrice $\widehat{\mathbf{L}}_s$. Plusieurs types de primitives \mathbf{s} ont été proposées auparavant. La plupart des travaux passés ont considéré des objets simples qui peuvent être exprimés par des primitives simples de type points, droites, ellipses, etc. Par conséquent, la famille des objets pouvant être traités par ce type de méthodes est limitée. Par ailleurs, ce genre de méthodes nécessite la mise en correspondance entre les primitives dans un couple d'images, ce qui n'est pas toujours facile à obtenir. D'autres méthodes ont été proposées pour surmonter les problèmes mentionnés auparavant en utilisant par exemple la projection dans l'espace propre [7] ou la signature du contour de l'objet [5]. Récemment, une nouvelle méthode utilisant les moments d'une image a été proposée dans [4]. Dans ce papier on propose une large amélioration de cette méthode.

Nous essayons en effet de déterminer quelles sont les primitives qui permettent d'avoir un comportement optimal du système. La première condition nécessaire à assurer est que la matrice d'interaction soit non singulière. Un bon choix de ces primitives doit donc permettre d'obtenir une matrice d'interaction \mathbf{L}_s de rang plein 6 dans un domaine le plus large possible. Un bon moyen pour assurer cette condition est de réaliser un schéma de commande découplée, c-à-d, pour chaque degré de liberté à commander, pouvoir extraire de l'image une caractéristique qui ne dépende que de ce degré de liberté. Dans ce cas de figure,

le jacobien des informations visuelles serait diagonal. Malheureusement ce découplage est idéal et certainement impossible à atteindre. Sans être aussi exigeant, nous pourrions nous contenter d'un jacobien diagonal par bloc ou triangulaire par bloc qui découplerait les rotations des translations. Ce découplage permettra de déterminer et d'éviter facilement les singularités. En pratique, il peut être obtenu en utilisant un asservissement 3D [19], mais ce genre de méthode nécessite une connaissance du modèle 3D de l'objet. Ce découplage peut être obtenu également en utilisant un asservissement visuel 2 1/2D [12], où la connaissance du modèle de l'objet n'est pas nécessaire. Par contre une matrice d'homographie doit être déterminée à chaque itération (ce qui rend cette méthode légèrement plus sensible aux erreurs de mesure dans l'image). En asservissement 2D, une première tentative a été proposée dernièrement dans [6, 4]. Dans cet article, une méthode plus efficace basée sur les moments invariants est proposée.

En asservissement visuel 2D, le comportement des primitives dans l'image est généralement satisfaisant. Cependant, la trajectoire du robot est généralement imprévisible et peu satisfaisante pour de grands déplacements en rotation [3]. En fait, la différence entre le comportement dans l'image et l'espace 3D est due au non-linéarités dans la matrice d'interaction. Pour expliquer cela, considérons la matrice d'interaction classique liée aux coordonnées d'un point (x, y) de l'image :

$$\mathbf{L}_x = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -1-x^2 & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} \quad (5)$$

On peut remarquer que la dynamique de \dot{x} et \dot{y} par rapport à chaque composante de la vitesse de la caméra n'est pas la même : quelques-unes sont inversement proportionnelles à la profondeur Z d'un point, certaines sont proportionnelles aux coordonnées de l'image du point et d'autres sont dépendantes du carré de ces coordonnées. Même s'il est possible de réaliser un schéma de commande tel que l'on obtienne une décroissance exponentielle découplée de l'erreur dans l'image, la dynamique du robot quant à elle sera très éloignée d'une décroissance exponentielle, du fait de la forte non linéarité de la matrice d'interaction \mathbf{L}_x . La trajectoire du robot sera donc loin d'une trajectoire optimale. Un problème important reste donc à résoudre : déterminer des primitives dont la matrice d'interaction associée est la moins non linéaire possible. Dans cet article, trois nouvelles primitives sont proposées pour commander les 3 ddl de translations. Elles sont telles que la matrice d'interaction associée est diagonale et constante par rapport aux trois ddl de translation. Deux nouvelles primitives sont aussi proposées en se basant sur les moments invariants par rapport aux mouvements de rotations, de translations et aux changements d'échelle pour commander les deux vitesses de rotation Ω_x et Ω_y . Dans la suite, nous donnons quelques définitions sur les moments et leurs propriétés. Dans la section 3 nous proposons six primitives pour contrôler les six ddl du robot. Finalement, notre schéma est validé par des résultats expérimentaux décrits dans la section 4.

2 Moments invariants

2.1 Définitions

Les moments 2D m_{pq} d'ordre $p + q$ de la fonction de densité $f(x, y)$ ($f(x, y)$ est le niveau de gris dans le cas d'une image) sont définis par :

$$m_{pq} \equiv \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) dx dy \quad (6)$$

Les moments centrés μ_{pq} sont calculés par rapport au centre de gravité de l'objet (x_g, y_g) . Ils sont calculés par :

$$\mu_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - x_g)^p (y - y_g)^q f(x, y) dx dy \quad (7)$$

où $x_g = \frac{m_{10}}{m_{00}}$ et $y_g = \frac{m_{01}}{m_{00}}$, $a = m_{00}$ est la surface de l'objet. Les moments centrés peuvent être calculés directement à partir des moments m_{pq} d'ordres inférieurs à $p + q + 1$ par :

$$\mu_{pq} = \sum_{k=0}^p \sum_{j=0}^q \binom{p}{k} \binom{q}{l} (-x_g)^{(p-k)} (-y_g)^{(q-l)} m_{kl} \quad (8)$$

Les moments d'une fonction de densité f existent si f est continue par morceau et n'est non nulle que sur des régions finies de l'espace. Les moments centrés sont invariants par rapport aux translations dans l'image. Les moments invariants par rapport aux rotations sont donnés en général sous formes polynômiales. Dans la littérature, plusieurs travaux proposent des méthodes de dérivation des moments invariants. Reddi [16] a obtenu des moments invariants par rapport aux rotations à partir des moments radiaux et angulaires, Teague [17], Belkassim [2], Walin et Kübler [18] ont déterminé des moments de Zernike invariants par rapport aux rotations, Abu-Mustapha et Psaltis [1], Flusser [9] ont obtenu des invariants par rapport aux rotations à partir des moments complexes. Finalement, plusieurs formules ont été proposées pour les invariants par rapport aux changements d'échelle, par exemple celle proposée dans [13] :

$$I = \frac{m_{pq}}{m_{00}^{(p+q+2)/2}} \quad (9)$$

Plus de détails sur les moments peuvent être trouvés dans [14] et [15]. Dans la suite, on présente quelques invariants par rapport aux rotations et translations dans l'image et par rapport aux changements d'échelle. Ils seront utilisés dans notre schéma de commande pour contrôler les vitesses Ω_x et Ω_y .

$$R_{l1} = \frac{In_1}{In_2}, R_2 = \frac{In_3}{In_2}, R_3 = \frac{In_4}{In_6}, R_4 = \frac{In_5}{In_6} \quad (10)$$

où

$$\begin{aligned} In_1 &= \mu_{30}^4 + 6\mu_{30}^3\mu_{12} + 9\mu_{30}^2\mu_{12}^2 + 6\mu_{30}^2\mu_{21}\mu_{03} \\ &+ 2\mu_{30}^2\mu_{03}^2 + 18\mu_{30}\mu_{21}\mu_{12}\mu_{03} \\ &+ 6\mu_{30}\mu_{12}\mu_{03}^2 + 9\mu_{21}^2\mu_{03}^2 + 6\mu_{21}\mu_{03}^3 + \mu_{03}^4 \end{aligned} \quad (11)$$

$$\begin{aligned}
I_{n_2} = & 3\mu_{30}^2\mu_{12}^2 + 2\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}^2\mu_{12} \\
& - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 2\mu_{30}\mu_{12}^3 + 3\mu_{21}^4 \\
& + 2\mu_{21}^3\mu_{03} + 3\mu_{21}^2\mu_{03}^2 - 6\mu_{21}\mu_{12}^2\mu_{03} + 3\mu_{12}^4
\end{aligned} \quad (12)$$

$$\begin{aligned}
I_{n_3} = & -\mu_{30}^2\mu_{03}^2 + 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} - 4\mu_{30}\mu_{12}^3 \\
& - 4\mu_{21}^3\mu_{03} + 3\mu_{21}^2\mu_{12}^2
\end{aligned} \quad (13)$$

$$I_{n_4} = (\mu_{50} + 2\mu_{32} + \mu_{14})^2 + (\mu_{05} + 2\mu_{23} + \mu_{41})^2 \quad (14)$$

$$I_{n_5} = (\mu_{50} - 2\mu_{32} - 3\mu_{14})^2 + (\mu_{05} - 2\mu_{23} - 3\mu_{41})^2 \quad (15)$$

$$I_{n_6} = (\mu_{50} - 10\mu_{32} + 5\mu_{14})^2 + (\mu_{05} - 10\mu_{23} + 5\mu_{41})^2 \quad (16)$$

2.2 Matrice d'interaction des moments 2D

Dans ce paragraphe, nous rappelons la matrice d'interaction des moments 2D donnée dans [4]. Dans ce qui suit nous supposons que l'objet appartient à un plan dont l'équation peut être déduite de :

$$\frac{1}{Z} = Ax + By + C$$

Nous supposons également que l'image est binaire ou que le niveau de gris ne change pas si la caméra bouge (c-à-d $\frac{df(x,y)}{dt} = 0$). Dans ce cas, la matrice d'interaction associée au moment m_{ij} peut être déterminée :

$$\mathbf{L}_{m_{ij}} = (m_{vx} \ m_{vy} \ m_{vz} \ m_{wx} \ m_{wy} \ m_{wz}) \quad (17)$$

où :

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

De la même façon, la matrice d'interaction associée au moment centré μ_{ij} est donnée par :

$$\mathbf{L}_{\mu_{ij}} = (\mu_{vx} \ \mu_{vy} \ \mu_{vz} \ \mu_{wx} \ \mu_{wy} \ \mu_{wz}) \quad (18)$$

avec :

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

où $n_{ij} = 4\mu_{ij}/m_{00}$. Pour les positions où l'objet est parallèle au plan image (c-à-d $A = B = 0$), on retrouve facilement à partir des deux premiers éléments de $\mathbf{L}_{\mu_{ij}}$ que les

variations des moments centrés par rapport aux vitesses V_X et V_Y s'annulent, ce qui prouve leurs invariance par rapport aux translations 2D parallèle au plan image si l'objet est parallèle au plan image. Pour les mêmes positions, on peut facilement prouver que les moments invariants par rapport aux changement d'échelle sont invariants par rapport aux translations suivant V_Z . Même si cette invariance est locale (c-à-d valable seulement si $A = B = 0$), ces primitives dépendent principalement des rotations. Elles seront donc utilisées pour commander les rotations.

3 Choix du vecteur des primitives

Dans cette section, nous choisissons six primitives pour commander les six degrés de liberté du robot en se basant sur les résultats théoriques présentés auparavant. Notre objectif est d'obtenir une matrice d'interaction creuse qui change doucement autour de la position désirée. La solution que nous présentons est telle que la matrice d'interaction est triangulaire si l'objet est parallèle au plan image. En outre, pour ces positions, les éléments de la matrice d'interaction qui correspondent aux translations forment un bloc diagonal constant. Dans [4] cette propriété n'était pas vérifiée. Dans la suite, nous supposons que la position désirée du plan image est parallèle à l'objet (c-à-d. $A = B = 0$) et on note \mathbf{L}_s^{\parallel} la matrice d'interaction qui correspond à cette position.

3.1 Primitives pour commander les translations

Dans [6, 4], les trois primitives proposées pour commander les 3 ddl de translations sont les coordonnées du centre de gravité (x_g, y_g) et la surface de l'objet a dans l'image. Dans ce cas, on obtient à partir de (17) :

$$\begin{aligned}
\mathbf{L}_{x_g}^{\parallel} &= (-C \ 0 \ Cx_g \ \epsilon_1 \ -(1+\epsilon_2) \ y_g) \\
\mathbf{L}_{y_g}^{\parallel} &= (0 \ -C \ Cy_g \ 1+\epsilon_3 \ -\epsilon_1 \ -x_g) \\
\mathbf{L}_a^{\parallel} &= (0 \ 0 \ 2aC \ 3ay_g \ -3ax_g \ 0)
\end{aligned} \quad (19)$$

avec $\epsilon_1 = n_{11} + x_g y_g$, $\epsilon_2 = n_{20} + x_g^2$ et $\epsilon_3 = n_{02} + y_g^2$. Même si cette matrice est triangulaire, on note la forte non linéarité de ses éléments. On peut aussi noter que les dynamiques ne sont pas les mêmes sur chaque axe de translation. Notre choix porte sur ces primitives intuitives mais en rajoutant une normalisation adéquate. Plus précisément, on définit :

$$a_n = Z^* \sqrt{\frac{a^*}{a}}, \quad x_n = a_n x_g, \quad y_n = a_n y_g \quad (20)$$

où a^* est la surface désirée de l'objet dans l'image et Z^* est la profondeur désirée entre le plan image de la caméra et l'objet. La matrice d'interaction associée à ces primitives normalisées peut être déterminée à partir de (19). Notons que $Z^* \sqrt{a^*} = Z \sqrt{a} = \sqrt{S}$ où S est la surface de l'objet

3D, on obtient :

$$\begin{aligned} \mathbf{L}_{x_n}^{\parallel} &= (-1 \ 0 \ 0 \ a_n \epsilon_{11} \ -a_n(1+\epsilon_{12}) \ y_n) \\ \mathbf{L}_{y_n}^{\parallel} &= (0 \ -1 \ 0 \ a_n(1+\epsilon_{21}) \ -a_n \epsilon_{11} \ -x_n) \\ \mathbf{L}_{a_n}^{\parallel} &= (0 \ 0 \ -1 \ -3y_n/2 \ 3x_n/2 \ 0) \end{aligned} \quad (21)$$

avec $\epsilon_{11} = n_{11} - x_g y_g / 2$, $\epsilon_{12} = n_{20} - x_g^2 / 2$, et $\epsilon_{21} = n_{02} - y_g^2 / 2$. Puisque a_n est inversement proportionnelle à \sqrt{a} , on retrouve le résultat récent présenté dans [11] qui montre qu'une telle primitive est proportionnelle à la profondeur (on note le terme constant dans le troisième élément de $\mathbf{L}_{a_n}^{\parallel}$). La normalisation par $Z^* \sqrt{a^*}$ n'a été faite que pour que ce terme constant soit égal à -1 . En outre, le choix de x_n et y_n permet un découplage total des 3 ddl de translation. On obtient aussi la même dynamique sur les trois axes de translation (notons le bloc diagonal égal à $-\mathbf{I}_3$ dans (21)). Cette bonne propriété permettra d'obtenir une bonne trajectoire des translations de la caméra. Finalement, on peut noter à partir des formules analytiques et des valeurs numériques des matrices d'interaction $\mathbf{L}_{x_n}^{\parallel}$ et $\mathbf{L}_{y_n}^{\parallel}$ (voir 25), le couplage classique entre V_X et Ω_Y et entre V_Y et Ω_X . Ce couplage naturel est utilisé pour que l'objet de reste dans le champ de vision de la caméra.

3.2 Primitives pour commander les rotations

Comme dans [6, 4], on utilise $\theta = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)$, défini comme angle d'orientation de l'axe principal d'inertie avec l'axe des abscisses du repère lié à l'image. On peut noter à partir de la figure 1 qu'il y a deux solutions pour θ : θ et $\theta + \pi$. Cependant, les moments d'ordre 3 peuvent être employés pour résoudre cette ambiguïté puisqu'une rotation d'un objet par π change le signe des moments d'ordre 3. En effet, pour distinguer les deux solutions de θ , on peut fixer le sens des axes principaux dans une image de référence. Le sens des axes principaux dans une image donnée est choisi tel que les moments d'ordre 3 calculés dans le repère défini par les axes principaux aient les mêmes signes que leurs correspondants calculés pour l'image de référence.

Nous employons également deux moments invariants R_i

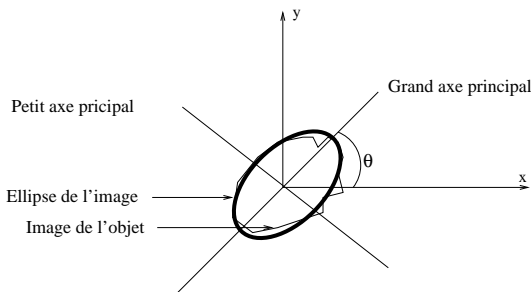


FIG. 1 – Ellipse d'un objet.

et R_j choisis parmi ceux donnés dans (10). La matrice d'interaction liée à ces invariants peut être obtenue à partir de

(18). On obtient alors (après des calculs fastidieux) :

$$\begin{aligned} \mathbf{L}_{R_i}^{\parallel} &= (0 \ 0 \ 0 \ R_{i_{wx}} \ R_{i_{wy}} \ 0) \\ \mathbf{L}_{R_j}^{\parallel} &= (0 \ 0 \ 0 \ R_{j_{wx}} \ R_{j_{wy}} \ 0) \\ \mathbf{L}_{\theta}^{\parallel} &= (0 \ 0 \ 0 \ \theta_{wx} \ \theta_{wy} \ -1) \end{aligned} \quad (22)$$

où les formes analytiques des éléments correspondant à Ω_x et à Ω_y ne sont pas données ici en raison de leur complexité (voir dans [4] les formules de θ_{wx} et θ_{wy}). Comme prévu, on peut noter l'invariance de ces primitives par rapport à tout mouvement de translation 3D (rappelons que nous considérons que $A = B = 0$) et l'invariance de R_i et R_j par rapport à Ω_Z . Finalement, on peut déduire de (25) que θ dépend essentiellement de la rotation Ω_z autour de l'axe optique. Quant à R_i et R_j , ils sont choisis tels que leurs matrices d'interaction associées $\mathbf{L}_{R_i}^{\parallel}$ et $\mathbf{L}_{R_j}^{\parallel}$ soient les plus orthogonales possible. Dans la section suivante, on présente des résultats expérimentaux où R_3 et R_4 ont été utilisés. Pour l'objet considéré, ce choix a donné les meilleurs résultats.

4 Résultats expérimentaux

Dans cette section, on présente quelques résultats expérimentaux obtenus avec un système robotique à six degrés de liberté. Les moments sont calculés à la cadence vidéo après une simple binarisation de l'image sans aucune segmentation spatiale. Comme indiqué, on utilise le vecteur suivant comme vecteur de primitives :

$$\mathbf{s} = (x_n \ y_n \ a_n \ R_3 \ R_4 \ \theta)^T \quad (23)$$

Dans nos expériences, les paramètres du plan de l'objet dans le repère caméra sont donnés approximativement pour la position désirée ($A = B = 0$, $C = 2$, qui correspond à $Z^* = 0.5$ m). Ils ne sont pas estimés à chaque itération. Pour les deux premières expériences, les valeurs correctes des paramètres intrinsèques de la caméra ont été utilisés. La valeur désirée \mathbf{s}^* est donnée par :

$$\mathbf{s}^* = (Z^* x_g^* \ Z^* y_g^* \ Z^* \ R_3^* \ R_4^* \ \theta^*)^T \quad (24)$$

où $x_g^*, y_g^*, R_3^*, R_4^*$ et θ^* sont calculés hors-ligne directement à partir de l'image désirée et Z^* a été fixé à 0.5 m. On peut déduire de (24), (23) et (20) que l'utilisation d'une valeur erronée \widehat{Z}^* de Z^* n'a pas d'effet sur la position de convergence du système ($\mathbf{s} = \mathbf{s}^*$ seulement pour la position désirée quelle que soit la valeur \widehat{Z}^*). Elle induit seulement un effet de gain (égal à \widehat{Z}^*/Z^*) sur la décroissance des trois premières primitives.

4.1 Mouvement de translation pure

Dans un premier temps, on compare les résultats obtenus en utilisant nos primitives et en utilisant les coordonnées du centre de gravité (x_g, y_g) et la surface a dans le cas où le déplacement est une translation pure entre la position initiale et la position désirée (données dans les figures 2.a

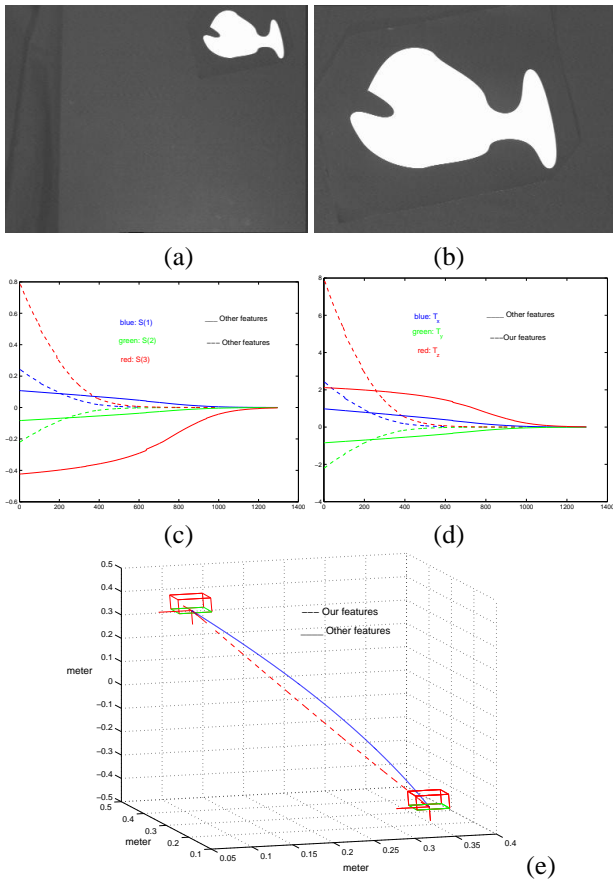


FIG. 2 – Résultats pour un mouvement de translation pure : (a) image initiale, (b) image désirée, (c) primitives visuelles ($\mathbf{s}-\mathbf{s}^*$), (d) vitesses de la caméra \mathbf{T}_c , (e) trajectoire 3D de la caméra

et 2.b). Pour les deux schémas de commande, la valeur de la matrice d'interaction calculée pour l'image désirée ($\widehat{\mathbf{L}}_{\mathbf{s}} = \mathbf{L}_{\mathbf{s}} | \mathbf{s} = \mathbf{s}^*$) a été utilisée dans la loi de commande (4) et la valeur du gain λ a été fixée à 0.1. La figure 2 montre les larges améliorations apportées par l'utilisation des primitives proposées dans ce papier (courbes en pointillé), puisqu'on obtient la même décroissance exponentielle découplée pour les primitives dans l'image et pour les composantes de la vitesse de la caméra. Comme prévu, la trajectoire obtenue en utilisant les primitives proposées est rectiligne, ce qui n'est pas le cas en utilisant les autres primitives.

4.2 Mouvement complexe

On teste maintenant notre schéma de commande pour des déplacements comportant des grandes translations et rotations à réaliser (voir figures 3.a et 3.b). La matrice d'inter-

action calculée pour la position désirée a la forme suivante:

$$\mathbf{L}_{\mathbf{s}|_{\mathbf{s}=\mathbf{s}^*}} = \begin{pmatrix} -1 & 0 & 0 & 0.01 & -0.52 & 0.01 \\ 0 & -1 & 0 & 0.51 & -0.01 & 0.01 \\ 0 & 0 & -1 & -0.02 & -0.01 & 0 \\ 0 & 0 & 0 & -0.33 & -0.62 & 0 \\ 0 & 0 & 0 & -0.61 & 0.09 & 0 \\ 0 & 0 & 0 & -0.04 & -0.08 & -1 \end{pmatrix} \quad (25)$$

On peut noter que cette matrice est bloc triangulaire avec des valeurs non nulles principalement sur la diagonale. La valeur du conditionnement de cette matrice (égal à 2,60) est également très satisfaisante. Enfin, nous avons utilisé la matrice $\widehat{\mathbf{L}}_{\mathbf{s}}$ suivante dans la loi de commande (4) :

$$\widehat{\mathbf{L}}_{\mathbf{s}} = \frac{1}{2}(\mathbf{L}_{\mathbf{s}} + \mathbf{L}_{\mathbf{s}|_{\mathbf{s}=\mathbf{s}^*}}) \quad (26)$$

Ce choix a donné les meilleurs résultats. Les résultats obtenus sont donnés sur la figure 3. Ils montrent le bon comportement de la loi de commande. D'abord, nous pouvons noter la convergence rapide vers la position désirée (alors que le système ne converge pas pour les six primitives proposées dans [4]). De plus, il n'y a aucune oscillation dans la décroissance des erreurs sur les primitives visuelles, il y a juste une petite oscillation pour uniquement deux composantes de la vitesse de la caméra (voir figure 3.d). Finalement, la trajectoire 3D de la caméra obtenue (voir figure 3.e) est très satisfaisante même si le déplacement à réaliser était très grand, ce qui était un problème majeur en asservissement visuel 2D.

4.3 Résultats avec une mauvaise calibration de la caméra et occultation d'objet

Nous examinons maintenant la robustesse de notre approche vis-à-vis d'une mauvaise calibration du système. Pour cela, des erreurs ont été ajoutées sur paramètres intrinsèques de la caméra (25% sur la focale et 20 pixels sur les coordonnées du point principal) et sur les paramètres du plan objet ($\widehat{Z}^* = 0.8m$ à la place de $Z^* = 0.5m$). On peut noter aussi que les conditions d'éclairage entre les position initiale et désirée (voir Figure 4.a et 4.b) sont différentes. Par ailleurs, une occultation d'une partie de l'objet dans la position initiale a été considérée. Les résultats obtenus sont donnés sur la figure 4. Nous pouvons noter que le système converge malgré les mauvaises conditions d'expérimentations. Une fois l'occultation terminée, le comportement du système devient similaire à celui obtenu dans les expérimentations précédentes, ce qui valide la robustesse de notre méthode vis-à-vis de ces erreurs.

5 Conclusion

Dans cet article, nous avons proposé une nouvelle approche d'asservissement visuel basée sur les moments d'un objet. Notre approche présente plusieurs avantages : il n'y a aucune contrainte sur la forme de l'objet et le modèle de l'objet n'est pas exigé. Les moments invariants ont été utilisés pour découpler les degrés de liberté de rotation de la caméra. Cela permet au système d'avoir un grand domaine

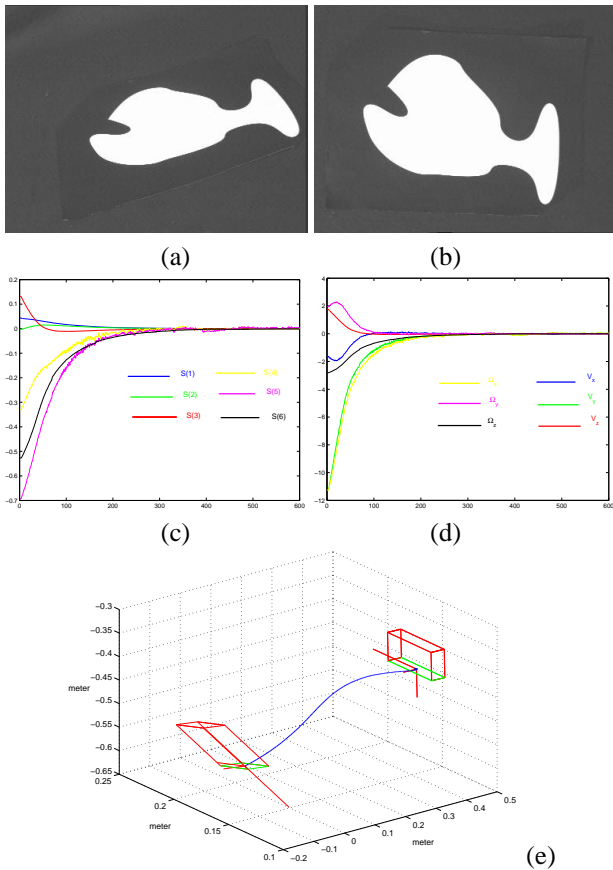


FIG. 3 – Résultats pour un mouvement complexe: (a) image initiale, (b) image désirée, (c) primitives visuelles ($\mathbf{s} - \mathbf{s}^*$), (d) vitesses de la caméra \mathbf{T}_c , (e) trajectoire 3D de la caméra

de convergence et un bon comportement à la fois des primitives dans l'image et pour la trajectoire de la caméra. Les résultats expérimentaux montrent la validité de notre approche et sa robustesse vis-à-vis des erreurs de calibration. Pour améliorer les résultats obtenus, nos travaux futurs vont porter sur le calcul de pose entre l'objet et la caméra afin de pouvoir mettre à jour la matrice d'interaction à chaque itération de la loi de commande.

Références

- [1] Y. S. Abu-Mustapha et D. Psaltis. Image normalisation by complex moments. *IEEE Trans. on PAMI*, 7(1):46–55, 1985.
- [2] S. O. Belkassim, M. Shridhar, et M. Ahmadi. Shape-contour recognition using moment invariants. In *10th Int. Conf. on Pattern Recognition*, pages 649–651, Atlantic City, NJ, USA, June 1990.
- [3] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In A.S. Morse D. Kriegman, G. Hager, editor, *The Confluence of Vision and Control*, number 237, pages 66–78. Springer-Verlag, 1998. LNCIS.
- [4] F. Chaumette. A first step toward visual servoing using image moments. In *IEEE/RSJ IROS'02*, pages 378–383,

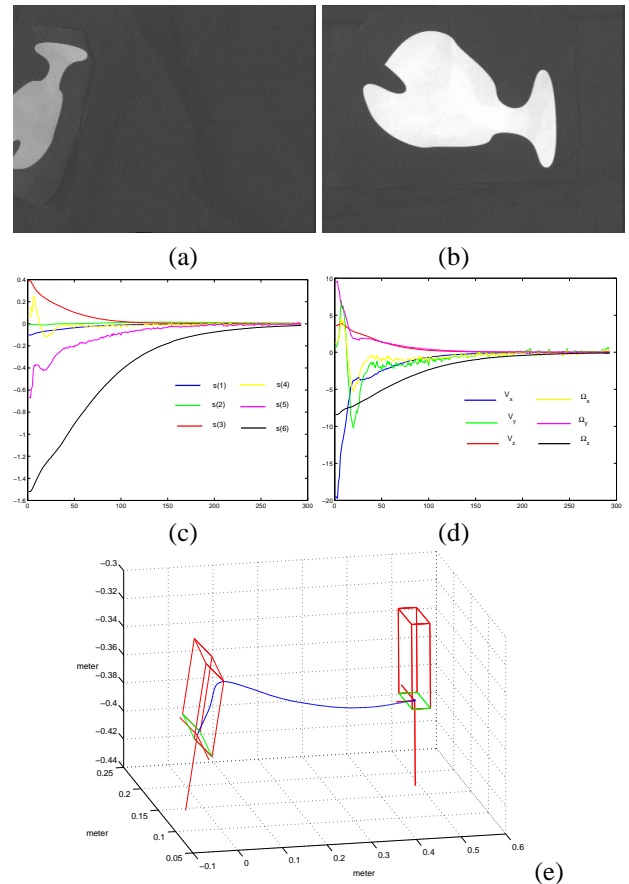


FIG. 4 – Résultats avec une mauvaise calibration: (a) image initiale, (b) image désirée (c) primitives visuelles ($\mathbf{s} - \mathbf{s}^*$), (d) vitesses de la caméra \mathbf{T}_c , (e) trajectoire 3D de la caméra

Lausanne, Switzerland, Oct. 2002.

- [5] C. Collewet et F. Chaumette. A contour approach for image-based control of objects with complex shape. In *IEEE/RSJ IROS'00*, pages 751–756, Takamatsu, Japan, Nov. 2000.
- [6] P. I. Corke et S. A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Trans. on Robotics and Automation*, 17(4):507–515, Aug. 2001.
- [7] K. Deguchi. A direct interpretation of dynamic images with camera and object motions for vision guided robot control. *Int. Journal of Computer Vision*, 37(1):7–20, 2000.
- [8] B. Espiau, F. Chaumette, et P. Rives. A new approach to visual servoing in robotics. In *IEEE Trans. on Robotics and Automation*, volume 8, pages 313–326, June 1992.
- [9] J. Flusser. On the independance of rotation moment invariants. *Pattern Recognition*, 33:1405–1410, 2000.
- [10] S. Hutchinson, G. Hager, et P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, Oct. 1996.
- [11] R. Mahony, P. Corke, et F. Chaumette. Choice of image features for depth-axis control in image-based visual servo control. In *IEEE/RSJ IROS'02*, pages 390–395, Lausanne, Switzerland, Oct. 2002.
- [12] E. Malis, F. Chaumette, et S. Boudet. 2 1/2 d visual servoing. *IEEE Trans. on Robotics and Automation*,

15(2):238–250, Apr. 1999.

- [13] A. G. Mamistvalov. n-dimensional moment invariants and conceptual theory of recognition n-dimensional solids. *IEEE Trans. on PAMI*, 20(8):819–831, Aug. 1998.
- [14] R. Mukundan et K. R. Ramakrishnan. *Moment Functions in Image Analysis Theory and Application*. World Scientific Publishing Co.Pte.Ltd, 1998.
- [15] R.J. Prokop et A. P. Reeves. A survey of moments based techniques for unoccluded object representation. *Graphical models and Image Processing*, 54(5):438–460, Sep. 1992.
- [16] S. S. Reddi. Radial and angular moment invariants for image identification. *IEEE Trans. on PAMI*, 3(2):240–242, 1981.
- [17] M.R. Teague. Image analysis via the general theory of moments. *Journal of Opt. Soc. of America*, 70:920–930, Aug. 1980.
- [18] A. Walin et O. Kübler. Complete sets of complex zernike moments invariants and the role of the pseudo-invariants. *IEEE Trans. on PAMI*, 17(11):1106–1110, Nov. 1995.
- [19] W. Wilson, C. Hulls, et G. Bell. Relative end-effector control using cartesian position-based visual servoing. *IEEE Trans. on Robotics and Automation*, 12(5):684–696, Oct. 1996.