



HAL
open science

Asservissement visuel basé sur une reconstruction 3D par vision dynamique

Ali Alhaj, Christophe Collewet, F. Chaumette

► **To cite this version:**

Ali Alhaj, Christophe Collewet, F. Chaumette. Asservissement visuel basé sur une reconstruction 3D par vision dynamique. 16èmes journées jeunes chercheurs en robotique, Villeurbanne, 16-17 septembre 2002, 2002, Villeurbanne, France. pp.6. hal-02580740

HAL Id: hal-02580740

<https://hal.inrae.fr/hal-02580740v1>

Submitted on 19 Sep 2023

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.

ASSERVISSEMENT VISUEL BASÉ SUR UNE RECONSTRUCTION 3D PAR VISION DYNAMIQUE

Ali Alhaj¹, Christophe Collewet¹, François Chaumette²

¹Cemagref, 17 avenue de Cucillé, 35 044 Rennes cedex

²IRISA / INRIA, Campus de Beaulieu, 35 042 Rennes cedex

ali.alhaj@cemagref.fr

christophe.collewet@cemagref.fr

francois.chaumette@irisa.fr

Résumé - Cet article traite de la réalisation de tâches de positionnement par asservissement visuel dans le cas où l'objet observé est plan et immobile. Par contre, aucune connaissance a priori sur sa forme n'est nécessaire. Pour ce faire, une phase de reconstruction 3D est préalablement réalisée, elle conduit à l'obtention des paramètres du plan de l'objet dans le repère de la caméra. Ce calcul repose sur la mesure du mouvement 2D dans une zone d'intérêt et sur la mesure de la vitesse de la caméra. Une fois les paramètres du plan suffisamment stables, un asservissement visuel permet de contrôler l'orientation de la caméra par rapport à l'objet et d'assurer qu'il reste dans le champ de vision de la caméra quelle que soit l'orientation finale désirée. Précisons que la phase de reconstruction 3D reste active pendant la phase d'asservissement pour améliorer la précision des paramètres et par suite l'obtention d'une faible erreur de positionnement. Des résultats expérimentaux valident l'approche proposée quant à son efficacité ou aux précisions obtenues.

Mots clé - asservissement visuel, reconstruction 3D, vision dynamique, objet inconnu.

1 Introduction

Les asservissements visuels 2D permettent la réalisation de tâches robotiques directement à partir d'informations visuelles acquises par une caméra. Ces informations sont ensuite comparées aux informations visuelles désirées, la consigne visuelle, obtenues en position souhaitée du porteur [9]. Toutefois, cette façon de faire s'avère problématique quand l'objet observé est inconnu puisque la consigne visuelle l'est alors aussi. Dans [19], les auteurs utilisent des mouvements spécifiques pour réaliser une tâche d'alignement sans connaissance précise de la consigne. Cependant, leur étude se restreint au cas des mouvements planaires. Dans [4], les auteurs proposent d'utiliser des informations visuelles de type dynamique permettant ainsi la réalisation d'une tâche de positionnement consistant à amener la caméra en face de l'objet observé, supposé plan et immobile. Toutefois, cette approche dite *d'asservissement visuel dynamique* (ou $\frac{d^2D}{dt^2}$) nécessite l'estimation des paramètres d'un modèle de mouvement [16] conduisant actuellement à des temps de calcul important et donc à une faible dynamique de commande du porteur. De plus, elle ne permet pas de spécifier une orientation quelconque en position désirée de la caméra. Ce cas de figure a été pris en compte dans [3] où des informations de type géométrique sont utilisées. En effet, l'approche repose sur la maximisation dans l'image de la surface d'un triangle constitué à partir de trois points caractéristiques extraits de l'image. Pour ce faire, trois tâches doivent être réalisées séquentiellement ce qui peut conduire à des durées excessives des tâches à réaliser.

L'approche décrite ici se propose de traiter le même

cas de figure, la réalisation d'une tâche de positionnement par rapport à un objet plan et immobile *a priori* inconnu et ce pour une orientation quelconque désirée de la caméra. L'objet étant inconnu, une phase de reconstruction 3D par vision dynamique est préalablement réalisée. Ce calcul repose comme dans [4] sur la mesure du mouvement 2D dans une zone d'intérêt mais nous l'utilisons ensuite pour remonter à la structure de l'objet. Cette façon de faire autorise, comme on le verra par la suite, à plus de souplesse dans l'élaboration de la loi de commande en permettant en particulier de contrôler l'orientation de la caméra afin que l'objet reste dans le champ de vision de la caméra quelle que soit l'orientation finale désirée.

Cet article est structuré comme suit. Tout d'abord, nous présentons dans la section 2 un rapide état de l'art des méthodes de reconstruction 3D par vision dynamique, nous formulons le problème en section 3 et décrivons l'obtention de la structure de l'objet en section 4. La section 5 détaille la loi de commande utilisée. Les résultats expérimentaux sont rapportés en section 6. La conclusion de ces travaux et quelques perspectives de recherche sont présentées en section 7.

2 État de l'art

Considérons un point P de l'objet observé, point décrit par $\underline{P} = (X, Y, Z)^T$ dans un repère lié à la caméra, l'axe Z coïncidant avec celui de l'axe optique. En considérant une focale unité, ce point se projette en p , décrit par $\underline{p} =$

$(x, y, 1)^T$, suivant :

$$\underline{p} = \frac{P}{Z} \quad (1)$$

qui permet d'établir la relation bien connue [5, 6] :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -1-x^2 & y \\ 0 & 1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} T_c \quad (2)$$

avec $T_c = (\underline{V}^T, \underline{\Omega}^T)^T$ où $\underline{V} = (V_x, V_y, V_z)^T$ et $\underline{\Omega} = (\Omega_x, \Omega_y, \Omega_z)^T$ représentent respectivement les vitesses de translation et de rotation de la caméra exprimées dans son repère. Cette relation peut se réécrire sous la forme :

$$\begin{cases} \dot{x} = \frac{-V_x + xV_z}{Z} + xy\Omega_x - (1+x^2)\Omega_y + y\Omega_z \\ \dot{y} = \frac{-V_y + yV_z}{Z} + (1+y^2)\Omega_x - xy\Omega_y - x\Omega_z \end{cases} \quad (3)$$

dans laquelle, seule la profondeur Z est inconnue en supposant T_c mesuré.

Diverses façons d'aboutir existent, elles reposent sur différentes approches d'appréhender \dot{x} et \dot{y} . La façon la plus immédiate est d'approximer les vitesses \dot{x} (\dot{y}) par $\frac{\Delta x}{\Delta t}$ ($\frac{\Delta y}{\Delta t}$) [18]. Cette méthode se révèle toutefois imprécise du fait des erreurs introduites par la quantification. Une façon d'améliorer grandement cette approche est d'imposer $\dot{x} = \dot{y} = 0$ [2]. Cela se fait simplement par asservissement visuel, il s'agit par conséquent ici de vision active. Une autre approche consiste à supposer que l'intensité du point observé reste constante au cours du mouvement. Cette hypothèse permet alors d'introduire la contrainte supplémentaire suivante, aussi bien connue [8] :

$$\dot{x}I_x + \dot{y}I_y + I_t = 0 \quad (4)$$

où I_x , I_y et I_t représentent les dérivées spatio-temporelles de l'image. On introduit alors \dot{x} et \dot{y} donnés par (3) dans (4) et il vient une expression de Z [7, 14] (notons que ces travaux traitent le cas plus général où T_c est supposé inconnu aussi). Cette approche, dite *directe*, nécessite un calcul précis des dérivées I_x , I_y et I_t et s'avère donc peu précise en pratique. Une autre façon de procéder est de modéliser localement la surface de l'objet autour de P qui fournit une expression de $1/Z$ en fonction de la paramétrisation choisie, que l'on introduit ensuite dans (3). Cette relation permet d'exhiber un modèle paramétrique du mouvement 2D, paramètres que l'on peut par ailleurs obtenir par une méthode de calcul de mouvement 2D, comme dans [16] par exemple. Il vient finalement une expression de la structure de l'objet [15] (là aussi le cas où T_c est inconnu est traité, une modélisation autour d'un second point est nécessaire). Ces méthodes sont dites *indirectes* dans la mesure où elles nécessitent un calcul intermédiaire du mouvement 2D.

Signalons que, dans notre contexte, l'approche décrite dans [12] n'est pas utilisable puisqu'elle nécessite au cours du mouvement l'image finale que nous ne connaissons pas.

3 Modélisation

Supposons l'objet plan, ou tout au moins plan dans un voisinage de P , il vient alors :

$$Z = A + BX + CY \quad (5)$$

que l'on peut réécrire en fonction de p grâce à (1) sous la forme :

$$\frac{1}{Z} = \alpha + \beta x + \gamma y \quad (6)$$

avec $\alpha = \frac{1}{A}$, $\beta = -\frac{B}{A}$ et $\gamma = -\frac{C}{A}$. Cette expression est ensuite introduite dans (3), on peut ainsi montrer que le mouvement 2D se modélise exactement par un modèle paramétrique à 8 paramètres [1]. En négligeant les termes du second ordre, il vient :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} a_0 + a_1x + a_2y \\ b_0 + b_1x + b_2y \end{pmatrix} \quad (7)$$

où :

$$\begin{cases} a_0 = -\alpha V_x - \Omega_y \\ a_1 = \alpha V_z - \beta V_x \\ a_2 = \Omega_z - \gamma V_x \\ b_0 = -\alpha V_y + \Omega_x \\ b_1 = -\Omega_z - \beta V_y \\ b_2 = \alpha V_z - \gamma V_y \end{cases} \quad (8)$$

que l'on peut écrire sous forme matricielle de la façon suivante :

$$M\Theta = \Gamma \quad (9)$$

avec :

$$M = \begin{pmatrix} -V_x & 0 & 0 \\ V_z & -V_x & 0 \\ 0 & 0 & -V_x \\ -V_y & 0 & 0 \\ 0 & -V_y & 0 \\ V_z & 0 & -V_y \end{pmatrix}, \Gamma = \begin{pmatrix} a_0 + \Omega_x \\ a_1 \\ a_2 - \Omega_z \\ b_0 - \Omega_x \\ b_1 + \Omega_z \\ b_2 \end{pmatrix} \quad (10)$$

et :

$$\Theta = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} \quad (11)$$

Si on suppose connue la vitesse de la caméra et les paramètres du modèle de mouvement 2D, on obtient alors facilement Θ en inversant (9) au sens des moindres carrés :

$$\Theta = (M^T M)^{-1} M^T \Gamma \quad (12)$$

Toutefois ce système ne s'inverse correctement que si la matrice $M^T M$ est bien conditionnée, c'est-à-dire si :

$$\min(\nu_i) > \nu_s, \quad i = 1 \dots 3 \quad (13)$$

où ν_i sont les valeurs propres de $M^T M$ et ν_s un seuil donné. Nous utiliserons cette propriété dans le paragraphe 6.

Les sections 4 et 5 montrent respectivement comment calculer les paramètres du modèle de mouvement et comment élaborer ensuite une loi de commande.

4 Estimation des paramètres du mouvement 2D

Les algorithmes permettant le calcul des paramètres du modèle donné par (7) sont relativement complexes et d'une mise en œuvre peu aisée. De plus, ils sont coûteux en temps de calcul. Aussi, il nous a paru plus judicieux

d'explorer une approche basée sur la mesure du déplacement dans l'image plutôt que sur le calcul direct de la vitesse. Cette approche consiste en somme à suivre le point p d'une image à l'autre. Nous rappelons ici l'approche classiquement utilisée [10, 17] et nous verrons que son degré de complexité n'excède pas celui de l'inversion d'une matrice de dimension 6.

Ici aussi on suppose que l'intensité lumineuse reste inchangée d'une image à l'autre :

$$I(x, y, t) = I(x + \dot{x}\Delta t, y + \dot{y}\Delta t, t + \Delta t) \quad (14)$$

où Δt représente la période d'acquisition des images, \dot{x} et \dot{y} étant modélisés par (7).

Du fait de la présence de bruit, (14) n'est généralement pas satisfaite. Il est alors plus judicieux de considérer ce problème dans le cadre d'une optimisation où il faut trouver les paramètres qui minimisent le résidu suivant :

$$\epsilon = \sum_W (I(x, y, t) - I(x + \dot{x}\Delta t, y + \dot{y}\Delta t, t + \Delta t))^2 \quad (15)$$

où W est une fenêtre d'analyse centrée en p .

En supposant Δt et les déplacements suffisamment faibles, on peut réaliser un développement de Taylor au premier ordre de $I(x + \dot{x}\Delta t, y + \dot{y}\Delta t, t + \Delta t)$ que l'on introduit dans (15). L'expression obtenue étant à présent linéaire en les paramètres recherchés, la minimisation du critère s'effectue finalement par la résolution d'un système linéaire donné en (16) sur la page suivante. L'inversion est réalisée en pratique en utilisant un schéma itératif de type Newton-Raphson pour tenir compte de l'approximation introduite par le développement en série de Taylor.

Pour garantir un bon fonctionnement de l'algorithme, p est choisi parmi des points d'intérêt extraits de la première image. On le choisit même comme le meilleur point d'intérêt au sens où il sera correctement suivi. Pour ce faire, la méthode décrite dans [11] est utilisée.

5 Loi de commande

Rappelons tout d'abord la tâche à réaliser. Il s'agit d'assurer une orientation désirée quelconque pour la caméra par rapport au plan π décrit par (5) et passant par P mais aussi d'assurer que P restera toujours dans le champ de vision de la caméra.

Exprimons pour ce faire la rotation de la caméra à effectuer par rapport à la position désirée. Θ étant connu, les paramètres du plan π s'expriment facilement et, par suite, la normale unitaire \underline{n}_c à π en P :

$$\underline{n}_c = \frac{1}{\sqrt{\alpha^2 + \beta^2 + \gamma^2}} (\beta, \gamma, \alpha)^T \quad (17)$$

on peut alors exprimer la rotation à effectuer sous la forme $\underline{u}\theta$ où \underline{u} représente l'axe de la rotation et θ l'angle de cette rotation (voir la figure 1) :

$$\underline{u} = \frac{\underline{Z} \wedge \underline{n}_c}{\|\underline{Z} \wedge \underline{n}_c\|} \quad (18)$$

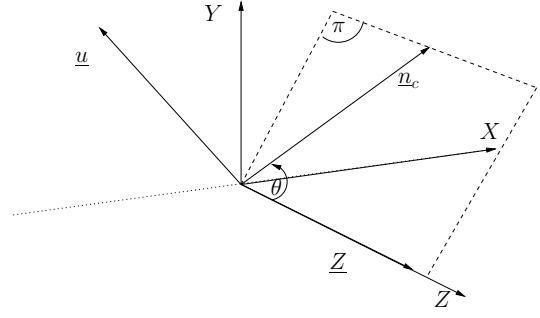


FIG. 1 – Rotation à réaliser par la caméra.

et :

$$\theta = \arccos(\underline{Z} \cdot \underline{n}_c) \quad (19)$$

\underline{Z} étant le vecteur unitaire porté par l'axe optique orienté vers P .

Notons qu'une orientation quelconque peut être atteinte en soumettant \underline{n}_c à la rotation désirée.

L'orientation de la caméra par rapport au plan π étant connue, l'élaboration d'une loi de commande est désormais possible. Nous avons utilisé celle décrite dans [13]. En effet, elle assure que le point P restera visible tout au long du mouvement en amenant en ligne droite p vers sa position souhaitée que nous avons choisie au centre du capteur. Nous rappelons ici brièvement cette approche, dite *d'asservissement visuel hybride*.

On définit \underline{p}_r de la façon suivante :

$$\underline{p}_r = \frac{1}{Z^*} \underline{P} = \frac{Z}{Z^*} \underline{p} \quad (20)$$

avec Z^* la profondeur souhaitée de P en position finale. On peut alors montrer que :

$$\dot{\underline{p}}_r = \left(-\frac{1}{Z^*} \mathbb{I}_3 \quad [\underline{p}_r]_{\times} \right) T_c \quad (21)$$

où la notation $[\underline{v}]_{\times}$ représente la matrice du pré-produit vectoriel associée à \underline{v} .

Le contrôle de l'orientation est quant à lui effectué de la façon suivante :

$$\frac{d(\underline{u}\theta)}{dt} = \begin{pmatrix} 0 & L_{\Omega} \end{pmatrix} T_c \quad (22)$$

avec :

$$L_{\Omega}(\underline{u}, \theta) = \mathbb{I}_3 - \frac{\theta}{2} [\underline{u}]_{\times} + \left(1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2\left(\frac{\theta}{2}\right)} \right) [\underline{u}]_{\times}^2 \quad (23)$$

où sinc représente la fonction sinus cardinal.

(21) et (22) peuvent se regrouper comme suit :

$$\frac{d}{dt} \begin{pmatrix} \underline{p}_r \\ \underline{u}\theta \end{pmatrix} = \begin{pmatrix} -\frac{1}{Z^*} \mathbb{I}_3 & [\underline{p}_r]_{\times} \\ 0 & L_{\Omega} \end{pmatrix} T_c = L T_c \quad (24)$$

Par ailleurs, la tâche de positionnement souhaitée peut s'exprimer en terme de régulation à zéro de la fonction de tâche suivante :

$$\underline{e} = \begin{pmatrix} \underline{p}_r - \underline{p}_r^* \\ \underline{u}\theta \end{pmatrix} \quad (25)$$

$$\begin{pmatrix} \sum I_x^2 & \sum xI_x^2 & \sum yI_x^2 & \sum I_xI_y & \sum xI_xI_y & \sum yI_xI_y \\ \sum xI_x^2 & \sum x^2I_x^2 & \sum xyI_x^2 & \sum xI_xI_y & \sum x^2I_xI_y & \sum xyI_xI_y \\ \sum yI_x^2 & \sum xyI_x^2 & \sum y^2I_x^2 & \sum yI_xI_y & \sum xyI_xI_y & \sum y^2I_xI_y \\ \sum I_xI_y & \sum xI_xI_y & \sum yI_xI_y & \sum I_y^2 & \sum xI_y^2 & \sum yI_y^2 \\ \sum xI_xI_y & \sum x^2I_xI_y & \sum xyI_xI_y & \sum xI_y^2 & \sum x^2I_y^2 & \sum xyI_y^2 \\ \sum yI_xI_y & \sum xyI_xI_y & \sum y^2I_xI_y & \sum yI_y^2 & \sum xyI_y^2 & \sum y^2I_y^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \end{pmatrix} = - \begin{pmatrix} \sum I_tI_x \\ \sum xI_tI_x \\ \sum yI_tI_x \\ \sum I_tI_y \\ \sum xI_tI_y \\ \sum yI_tI_y \end{pmatrix} \quad (16)$$

\underline{p}^* étant le centre du capteur.

La loi de commande permettant une convergence exponentielle de \underline{e} s'écrit donc :

$$T_c = -\lambda \hat{L}^{-1} \underline{e} \quad (26)$$

où λ est un gain positif réglant la vitesse de convergence.

Signalons que les valeurs de Z et de \underline{p} nécessaires au calcul de \hat{L} sont obtenues respectivement grâce à (5) et à l'intégration de (7).

6 Résultats expérimentaux

Afin de valider l'algorithme proposé nous présentons ici des résultats expérimentaux pour trois orientations différentes souhaitées. La plate-forme expérimentale utilisée est décrite dans [3].

L'objet est une photographie disposée sur un support plan. Ce support permet par ailleurs d'exprimer précisément la matrice de passage décrivant la rotation du repère de la caméra par rapport à celui de l'objet, et ce, grâce à la méthode utilisée dans [3]. Cette matrice est caractérisée par les angles d'Euler notés ϕ_X, ϕ_Y, ϕ_Z et décrivant respectivement les rotations autour des axes X, Y et Z .

Par ailleurs, l'objet étant fixe, on peut améliorer la précision sur Θ . En effet, en passant dans un repère fixe, on peut exprimer à chaque instant un Θ^f que l'on peut filtrer en remarquant que l'on cherche une valeur Θ_c^f constante. Cette valeur est exprimée ensuite dans le repère de la caméra pour être utilisée dans la loi de commande. En outre, cette façon de faire permet de contrôler l'instant où Θ_c^f peut être considéré comme suffisamment stable pour être utilisé dans la loi de commande (typiquement une quinzaine d'acquisitions s'avère suffisante). De ce fait, une phase initiale à vitesse constante est utilisée. Nous avons imposé $V_x = V_y = 2$ cm/s, $V_z = 0$ et $\underline{\Omega} = 0$.

Finalement, l'algorithme se décompose en trois phases, une première phase à vitesse constante, une seconde où, à la fois, la reconstruction et l'asservissement sont réalisés et une dernière phase où seul opère l'asservissement. Cette dernière phase correspond au cas où la contrainte (13) n'est plus satisfaite.

Enfin, toutes les expérimentations ont été réalisées dans les conditions suivantes : $\lambda = 0.06$, $\nu_s = 1.10^{-6}$, $\Delta t = 800$ ms, $Z^* = 70$ cm et la taille de W a été fixée à 55×55 .

La première expérimentation concerne la réalisation d'une tâche de positionnement en face de l'objet. La figure 2 rapporte le comportement de l'algorithme. Plus précisément, les figures 2.a-b décrivent respectivement l'évolution

des paramètres A, B et C du plan dans un repère fixe et dans le repère de la caméra, les figures 2.c-d l'évolution des paramètres du modèle de mouvement, respectivement a_0, b_0 et a_i, b_i ($i = 1, 2$), l'évolution du torseur cinématique sur la figure 2.e, l'erreur normalisée sur la figure 2.f¹, la profondeur estimée sur la figure 2.g. Finalement, les images initiale et finale sont rapportées respectivement sur les figures 2.h-i.

Tout d'abord, la figure 2.f confirme que la loi de commande converge puisque l'erreur normalisée tend vers zéro, tout comme \hat{Z} (figure 2.g) vers Z^* . De même, le point p tend vers le centre du capteur, comme prévu, en ligne droite (voir la figure 2.i où la trajectoire de p est dessinée, le premier segment de droite correspond au mouvement relatif à la phase initiale). Par ailleurs, on peut remarquer sur la figure 2.a que la reconstruction s'arrête vers la 120^{ème} itération, par contre, le critère d'arrêt sur l'erreur normalisée n'est atteint qu'au bout de 300 itérations environ. Pour cette expérimentation, l'orientation initiale de la caméra était $\phi_X = 9.66^\circ$, $\phi_Y = 23.77^\circ$ et $\phi_Z = 9.46^\circ$, l'orientation après asservissement devient $\phi_X = 2.44^\circ$ et $\phi_Y = 1.39^\circ$ (rappelons que ϕ_Z n'est pas contrôlé). Par conséquent, le positionnement est relativement précis.

La seconde expérimentation concerne cette fois une orientation désirée différente : $\phi_X = 0^\circ$ et $\phi_Y = 20^\circ$. Le comportement de l'algorithme est rapporté sur la figure 3. Les figures 3.a-i décrivent les mêmes variables que l'expérimentation précédente. Là aussi les mêmes commentaires peuvent être faits, en particulier sur la convergence de la loi de commande. Toutefois, comme le rapporte la figure 3.i, la convergence de p n'est plus en ligne droite. Vraisemblablement, il s'agit d'un mauvais comportement de l'algorithme de suivi. Cela est sans conséquence puisque l'erreur de positionnement est faible. En effet, l'orientation finale mesurée est $\phi_X = -0.65^\circ$ et $\phi_Y = 20.16^\circ$, l'orientation initiale était $\phi_X = 16.88^\circ$, $\phi_Y = 11.22^\circ$ et $\phi_Z = 7.40^\circ$.

La dernière expérimentation concerne une autre orientation désirée : $\phi_X = -10^\circ$ et $\phi_Y = 0^\circ$. Les figures 4.a-i décrivent les mêmes variables que précédemment. Cette expérimentation ne nécessite pas de commentaire particulier, le comportement de l'algorithme étant du même type que pour les expérimentations précédentes. L'erreur de positionnement est, ici aussi, faible puisque l'orientation finale est $\phi_X = -8.90^\circ$ et $\phi_Y = -0.06^\circ$, l'orientation initiale étant $\phi_X = 6.58^\circ$, $\phi_Y = -10.94^\circ$ et $\phi_Z = 2.83^\circ$.

¹définie comme étant $\frac{\|\underline{e}\|_{t=k\Delta t}}{\|\underline{e}\|_{t=0}}$.

7 Conclusion et perspectives

Nous avons présenté dans cet article une façon de réaliser une tâche de positionnement par asservissement visuel dans le cas où l'objet considéré est plan et immobile, par contre nous le supposons inconnu. L'approche est basée sur une reconstruction 3D par vision dynamique qui permet le calcul de l'orientation courante de la caméra utile à la loi de commande. L'algorithme proposé permet de prendre en compte des orientations quelconques désirées de la caméra. Il a été validé par des expérimentations. En effet, des faibles erreurs d'orientation ont pu être observées ($\approx 2^\circ$). Regrettons toutefois qu'il est sensible à la calibration du porteur comme à celle de la caméra.

Une perspective intéressante à ce travail concerne l'optimisation de la durée d'exécution de la tâche vis-à-vis de la trajectoire de la caméra en garantissant un mouvement dans π . Une autre perspective, tout aussi intéressante, est l'extension de l'algorithme au cas des objets non plans. En considérant une modélisation paramétrique locale en P l'algorithme ne devrait pas être profondément modifié.

Références

- [1] G. ADIV. « Determining 3D Motion and Structure From Optical Flow Generated by Several Moving Objects ». *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(4) :384–401, Juillet 1985.
- [2] F. CHAUMETTE, S. BOUKIR, P. BOUTHEMY, et D. JUVIN. « Structure from controlled motion ». *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(5) :492–504, Mai 1996.
- [3] C. COLLEWET et F. CHAUMETTE. « Positioning a Camera With Respect to Planar Objects of Unknown Shape by Coupling 2-D Visual Servoing and 3-D Estimations ». *IEEE Trans. on Robotics and Automation*, 18(3) :322–333, Juin 2002.
- [4] A. CRÉTUAL et F. CHAUMETTE. « Visual servoing based on image motion ». *Int. Journal of Robotics Research*, 20(11) :857–877, Novembre 2001.
- [5] B. ESPIAU, F. CHAUMETTE, et P. RIVES. « A new approach to visual servoing in robotics ». *IEEE Trans. on Robotics and Automation*, 8(3) :313–326, Juin 1992.
- [6] J. T. FEDDEMA et O. R. MITCHELL. « Vision-guided servoing with feature-based trajectory generation ». *IEEE Trans. on Robotics and Automation*, 5(5) :691–700, Octobre 1989.
- [7] B. K. P. HORN et E. J. WELDON. « Direct Methods for Recovering Motion ». *Int. Journal of Computer Vision*, 2(1) :51–76, Juin 1988.
- [8] B.K.P. HORN et B.G. SCHUNCK. « Determining Optical Flow ». *Artificial Intelligence*, 16(1–3) :185–203, Août 1981.
- [9] S. HUTCHINSON, G. D. HAGER, et P. I. CORKE. « A tutorial on visual servo control ». *IEEE Trans. on Robotics and Automation*, 12(5) :651–670, Octobre 1996.
- [10] M. IRANI, B. ROUSSO, et S. PELEG. « Recovery of Ego-Motion Using Region Alignment ». *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(3) :268–272, Mars 1997.
- [11] C. KERMADE, C. COLLEWET, et K. CHEHDI. « Apport de la sélection de primitives point à un suivi robuste ». Dans *13ème Congrès de Reconnaissance des Formes et d'Intelligence Artificielle, RFIA'2002*, vol. 2, p. 539–548, Angers, 8–10 Janvier, 2002.
- [12] E. MALIS et F. CHAUMETTE. « 2 1/2D Visual Servoing with Respect to Unknown Objects Through a New Estimation Scheme of Camera Displacement ». *Int. Journal of Computer Vision*, 37(1) :79–97, 2000.
- [13] E. MALIS et F. CHAUMETTE. « Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods ». *IEEE Trans. on Robotics and Automation*, 18(2) :176–186, Avril 2002.
- [14] S. NEGAHDARIPOUR et B. K. P. HORN. « Direct Passive Navigation ». *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(1) :168–176, Janvier 1987.
- [15] S. NEGAHDARIPOUR et S. LEE. « Motion Recovery from Image Sequences Using Only First Order Optical Flow Information ». *Int. Journal of Computer Vision*, 9(3) :163–184, 1992.
- [16] J.M. ODOBEZ et P. BOUTHEMY. « Robust multi-resolution estimation of parametric motion models ». *Journal of Visual Communication and Image Representation*, 6(4) :348–365, Décembre 1995.
- [17] J. SHI et C. TOMASI. « Good features to track ». Dans *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, p. 593–600, Seattle, Juin 1994.
- [18] M. XIE. « Contribution à la vision dynamique : reconstruction d'objets 3D polyédriques par une caméra mobile ». Thèse de doctorat, Université de Rennes I, Rennes, Juin 1989.
- [19] B. YOSHIMI et P. K. ALLEN. « Active uncalibrated visual servoing ». Dans *IEEE Int. Conf. on Robotics and Automation, ICRA'94*, p. 156–161, San Diego, Mai 1994.

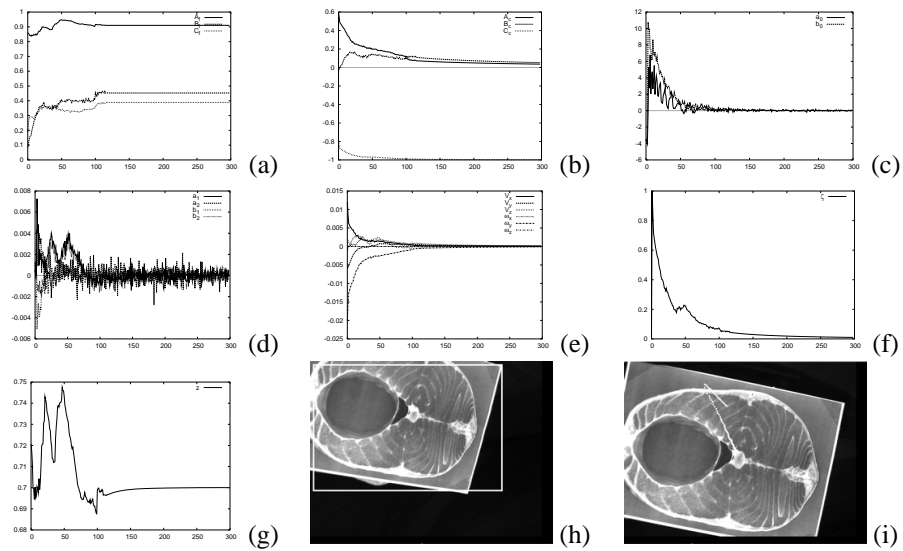


FIG. 2 – 1^{ère} expérimentation : (a) Paramètres du plan dans un repère fixe. (b) Paramètres du plan dans le repère de la caméra. (c) Paramètres a_0, b_0 . (d) Paramètres a_1, b_1, a_2, b_2 . (e) Torseur cinématique. (f) Erreur normalisée. (g) Profondeur estimée \hat{Z} . (h) Position initiale. (i) Position finale.

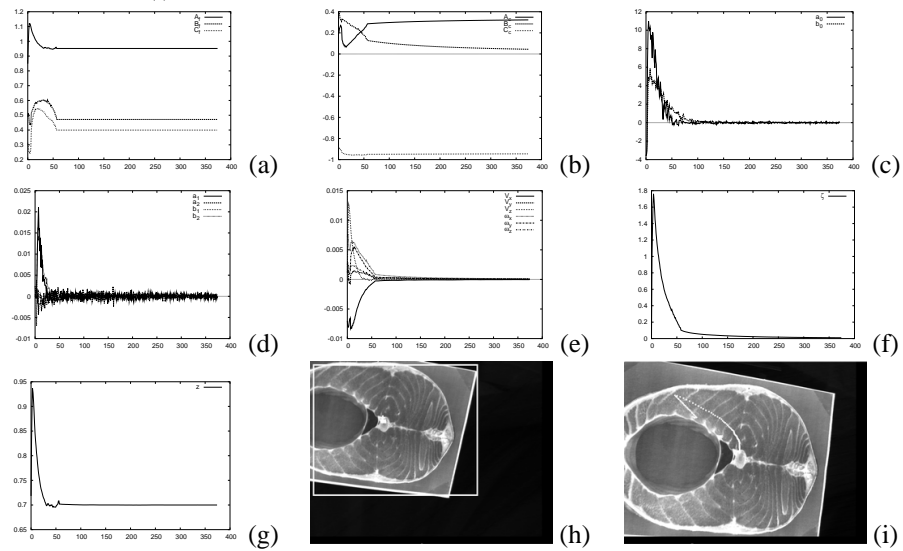


FIG. 3 – 2^{ème} expérimentation.

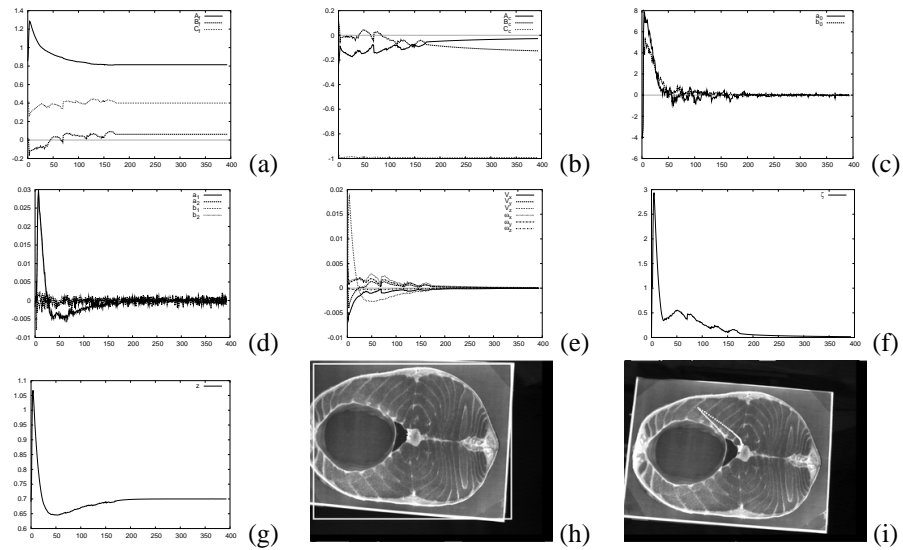


FIG. 4 – 3^{ème} expérimentation.