



**HAL**  
open science

# Couplage entre asservissement visuel et reconstruction 3D par vision dynamique

Alhaj Ali, Collewet Christophe, Chaumette François

► **To cite this version:**

Alhaj Ali, Collewet Christophe, Chaumette François. Couplage entre asservissement visuel et reconstruction 3D par vision dynamique. 19ème colloque GRETSI sur le traitement du signal et des images, GRETSI, Sep 2003, Paris, France. hal-04236831

**HAL Id: hal-04236831**

**<https://hal.inrae.fr/hal-04236831v1>**

Submitted on 11 Oct 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.

# Couplage entre asservissement visuel et reconstruction

## 3D par vision dynamique

Ali ALHAJ<sup>1</sup>, Christophe COLLEWET<sup>1</sup>, François CHAUMETTE<sup>2</sup>

<sup>1</sup>Cemagref

17 avenue de Cucillé, 35 044 Rennes, France

<sup>2</sup>IRISA/INRIA

Campus de Beaulieu, 35 042 Rennes, France

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. 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 déplacement 2D d’une zone d’intérêt et sur la mesure de la vitesse de la caméra. Une fois les paramètres du plan obtenus, un asservissement visuel permet de contrôler l’orientation de la caméra par rapport à l’objet et d’assurer qu’il reste dans son champ de vision quelle que soit l’orientation finale désirée. Précisons que la phase de reconstruction reste active pendant l’asservissement afin d’améliorer la précision des paramètres. Des résultats expérimentaux valident l’approche proposée quant à son efficacité et les précisions obtenues.

**Abstract** – We present a way to achieve positioning tasks by visual servoing in the case of planar and motionless objects whose shape is unknown. Emphasis is made on the algorithm of 3d reconstruction. More precisely, the reconstruction phase is based on the measurement of the 2D displacements in a region of interest and on the measurement of the camera velocity. Once the parameters of the plane are known, a visual servoing scheme is used to control the orientation of the camera with respect to the object and to ensure that it remains in the camera field of view for any desired orientation. The 3D reconstruction phase is maintained active during the servoing to improve the accuracy of the parameters. Experimental results validate the approach.

## 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 [7]. Toutefois, cette façon de faire s’avère problématique quand l’objet observé est inconnu puisque la consigne visuelle l’est alors aussi ; cela est le cas des objets que l’on rencontre en agroalimentaire, objets que l’on se propose de traiter ici. Pour ce faire, certains auteurs utilisent des mouvements spécifiques pour réaliser une tâche d’alignement sans connaissance précise de la consigne [11]. Cependant, leur étude se restreint au cas de mouvements planaires. Dans [3], 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 nécessite l’estimation des paramètres d’un modèle de mouvement 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 [2] où des informations de type géométrique sont utilisées. Toutefois, trois tâches doivent être réalisées séquentiellement ce qui peut conduire à une durée excessive de la tâche à réaliser. Une approche basée

sur un suivi robuste est proposée dans [4] permettant l’obtention d’une reconstruction projective. Malheureusement, les auteurs ne fournissent pas de résultats concernant la réalisation de tâches robotiques.

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 sera préalablement réalisée. On verra que cette façon de procéder conduit à plus de souplesse dans l’élaboration de la loi de commande en permettant en particulier le contrôle de l’orientation de la caméra afin que l’objet reste dans le champ de vision de la caméra.

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 décrivons ensuite l’obtention de la structure de l’objet en section 3. La section 4 décrit la loi de commande utilisée. Les résultats expérimentaux sont rapportés en section 5.

## 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. 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 [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.

Dans (2), si  $T_c$  est supposé mesuré, seul  $Z$  est inconnu. Diverses façons d'aboutir existent, elles reposent sur différentes approches d'appréhender  $\dot{x}$  et  $\dot{y}$ . Toutefois, nous nous contenterons ici de ne rappeler que les méthodes dites *indirectes*. Ce type d'approche repose sur une modélisation locale de la surface de l'objet autour de  $P$  permettant l'obtention de  $1/Z$  en fonction de la paramétrisation choisie, que l'on introduit ensuite dans (2). 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. Il vient finalement une expression de la structure de l'objet [9] (notons que ces travaux traitent le cas plus général où  $T_c$  est supposé inconnu aussi). Ces méthodes sont qualifiées d'indirectes dans la mesure où elles nécessitent un calcul intermédiaire du mouvement 2D. De telles approches, y compris notre approche précédente [1], supposent implicitement que la fréquence d'acquisition des images soit suffisamment élevée (ou la vitesse 3D de la caméra faible) afin de pouvoir considérer les paramètres du mouvement constants entre deux images successives.

Le principal acquis de l'approche proposée ici est de lever cette contrainte aussi bien pour le calcul de la structure de l'objet que pour le calcul du déplacement 2D. De ce fait, des vitesses élevées de la caméra peuvent être atteintes. De plus, l'utilisation d'une reconstruction 3D permet d'élaborer plus facilement la loi de commande en permettant en particulier le contrôle de l'orientation de la caméra quelle que soit l'orientation finale désirée.

### 3 Structure de l'objet

Supposons tout d'abord l'objet plan, ou tout au moins plan dans un voisinage de  $P$ , il vient alors à l'instant  $k$  :

$$\frac{1}{Z_k} = \alpha_k x_k + \beta_k y_k + \gamma_k = \underline{\Theta}_k^T \cdot \underline{p}_k \quad (3)$$

Par ailleurs, la caméra étant soumise à la vitesse  $T_c$ , on peut montrer que  $P_{k+1}$  vérifie :

$$\underline{P}_{k+1} = \mathcal{W}_k \underline{P}_k + \underline{\mathcal{V}}_k \quad (4)$$

où  $\mathcal{W}_k$  est une matrice de rotation dépendant de  $\underline{\Omega}$  et  $\underline{\mathcal{V}}_k$  un vecteur de translation dépendant de  $\underline{\Omega}$  et de  $\underline{V}$ . Par suite, en utilisant (3) puis (4) dans (1), on peut montrer le résultat bien connu selon lequel la projection de  $P_{k+1}$  dans l'image peut s'approximer suivant un modèle affine de déplacement :

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = M \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \underline{d} \quad (5)$$

où  $M$  et  $\underline{d}$  sont des fonctions de  $T_c$  et de  $\underline{\Theta}_k$ . Par conséquent, connaissant  $T_c$  et une mesure du déplacement entre les images

$k$  et  $k+1$ , on peut obtenir la structure de l'objet  $\underline{\Theta}_k$ . En effet, on peut montrer que (5) se met sous la forme :

$$\mathcal{M} \underline{\Theta}_k = \underline{\Gamma} \quad (6)$$

où  $\mathcal{M}$  est une matrice  $6 \times 6$  et  $\underline{\Gamma}$  un vecteur de dimension 6. Il vient alors une expression de la structure recherchée :

$$\underline{\Theta}_k = (\mathcal{M}^T \mathcal{M})^{-1} \mathcal{M}^T \underline{\Gamma}. \quad (7)$$

Il reste maintenant à réaliser une mesure du déplacement entre les images  $k$  et  $k+1$ . Pour ce faire, considérons deux images consécutives  $f$  et  $g$  et supposons que l'intensité demeure inchangée entre ces deux images :

$$f(\underline{p}_k) = g(\delta(\underline{p}_k)) \quad (8)$$

où  $\delta(\cdot)$  représente la déformation de l'image entre les instants  $k$  et  $k+1$  suivant (5). Du fait de la présence de bruit, (8) n'est généralement pas satisfaite. Il est alors plus judicieux de chercher les paramètres minimisant :

$$\epsilon = \sum_{p_l \in W} \left( f(\underline{p}_l) - g(\delta(\underline{p}_l)) \right)^2 \quad (9)$$

où  $W$  est une fenêtre d'analyse centrée en  $p_k$ .

L'approche classique consiste à supposer les déformations suffisamment petites pour rendre le problème linéaire [10]. Toutefois, comme la méthode de reconstruction permet de considérer des vitesses importantes, nous ne ferons pas cette hypothèse. Différentes façons de procéder sont alors possibles comme adopter des approches multi-échelle [4] ou multi-résolution [5]. Cependant, nous décrivons ici une approche moins lourde en temps de calcul.

En effet, comme  $T_c$  peut être mesuré, une estimation de  $\delta(\cdot)$  peut être obtenue grâce à (5) (une fois  $\underline{\Theta}_k$  connu, cette valeur peut être introduite dans (5) pour améliorer  $\widehat{M}$  et  $\widehat{\underline{d}}$ , sinon une approximation grossière de  $\underline{\Theta}_k$  est utilisée). Cette façon de faire peut être utilisée dès lors qu'un mouvement contrôlé est utilisé, typiquement en asservissement visuel ou en vision active. Par suite, en écrivant  $M$  et  $\underline{d}$  sous la forme  $M = \widehat{M} + \epsilon$  et  $\underline{d} = \widehat{\underline{d}} + \underline{\tau}$ , où  $\epsilon$  est une matrice de correction et  $\underline{\tau}$  également un vecteur de correction, et en introduisant  $\underline{\varsigma} = (\epsilon_{xx}, \epsilon_{yx}, \epsilon_{xy}, \epsilon_{yy}, \tau_x, \tau_y)^T$ , il devient possible de réaliser un développement en série de Taylor de  $g(\delta(\underline{p}_k))$  en supposant  $\underline{\varsigma}$  faible. On opère ensuite classiquement [10], soit tous calculs faits :

$$\underline{\varsigma}_{l+1} = \underline{\varsigma}_l + \eta \left( \sum_W \underline{\Phi}_l \underline{\Phi}_l^T \right)^{-1} \left( f(\underline{p}_k) - g(\delta(\underline{p}_l)) \right) \cdot \underline{\Phi}_l \quad (10)$$

avec  $\eta$  un pas de descente,  $\underline{\Phi}_l = (g_x, g_y, xg_x, xg_y, yg_x, yg_y)^T$  où  $g_x$  et  $g_y$  sont les dérivées temporelles de  $g$  en  $\widehat{\underline{p}}_{k+1}$ .

$\widehat{\underline{\Theta}}_k$  étant obtenu, intéressons nous maintenant à l'élaboration de la loi de commande.

### 4 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  $\pi$  décrit par (3) et passant par  $P$  tout en assurant qu'il restera dans le champ de vision de la caméra.

Une fois  $\underline{\Theta}$  connu, une expression du vecteur normal unitaire  $\underline{n}$  au plan  $\pi$  en  $P$  peut être obtenu dans le repère de la caméra.

Toutefois, dans le cas d'une orientation désirée quelconque, il est préférable de considérer  $\underline{n}^* = \mathcal{R}\underline{n}$  où  $\mathcal{R}$  est une matrice de rotation calculée à partir de l'orientation désirée (voir Figure 1). De ce fait, la loi de commande doit conduire à  $\underline{Z} = \underline{n}_c$  où  $\underline{Z}$  est le vecteur unitaire porté par l'axe optique et  $\underline{n}_c = -\underline{n}^*$ . Cette rotation à réaliser peut se mettre aisément sous la forme  $\underline{u}\theta$  où  $\underline{u}$  représente l'axe de la rotation et  $\theta$  l'angle de cette rotation :

$$\underline{u} = \frac{\underline{Z} \wedge \underline{n}_c}{\|\underline{Z} \wedge \underline{n}_c\|} \quad (11)$$

et

$$\theta = \arccos(\underline{Z} \cdot \underline{n}_c) \quad (12)$$

L'orientation de la caméra par rapport au plan  $\pi$  étant connue, l'élaboration d'une loi de commande est désormais possible. Nous avons utilisé celle décrite dans [8]. 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. Cette approche est basée sur la régulation à zéro de la fonction de tâche suivante :

$$\underline{e} = \left( \begin{array}{c} \frac{Z}{Z^*} p - p^* \\ \underline{u}\theta \end{array} \right) \quad (13)$$

où  $Z^*$  et  $p^*$  sont les valeurs désirées de  $Z$  et  $p$ .

La régulation de  $\underline{e}$  conduit à la loi de commande suivante :

$$T_c = -\lambda \hat{L}^{-1} \underline{e} \quad (14)$$

$\lambda$  étant un gain positif et  $\hat{L}$  une approximation de la matrice d'interaction donnée dans [1]. Elle dépend de  $\underline{u}$ ,  $\theta$ ,  $Z^*$ ,  $Z$  et  $\underline{p}$  ( $Z$  et  $\underline{p}$  étant obtenus grâce à (3) et (5)).

## 5 Résultats expérimentaux

Afin de valider l'algorithme proposé nous présentons ici des résultats expérimentaux pour deux orientations différentes souhaitées. La plate-forme expérimentale utilisée est décrite dans [2] (excepté le PC qui est un Pentium 4 à 2 Ghz), tout comme le dispositif permettant la vérification de l'orientation réellement atteinte, orientation caractérisée par les angles d'Euler notés  $\phi_X$ ,  $\phi_Y$ ,  $\phi_Z$  et décrivant respectivement les rotations autour des axes  $X$ ,  $Y$  et  $Z$ .

Par ailleurs, l'objet étant fixe, on peut améliorer la précision sur  $\underline{\Theta}$ . En effet, en passant dans un repère fixe, on peut exprimer à chaque instant un  $\underline{\Theta}^f$  que l'on peut filtrer en remarquant que l'on cherche une valeur  $\underline{\Theta}_c^f$  constante. En outre, cette façon de faire permet de contrôler l'instant où  $\underline{\Theta}_c^f$  peut être considéré comme stable pour être utilisé dans la loi de commande (typiquement une dizaine d'acquisitions s'avère suffisante). De ce fait, une phase initiale à vitesse constante est nécessaire au cours de laquelle nous avons imposé  $\|\underline{V}\| = 4$  cm/s,  $V_z = 0$  et  $\underline{\Omega} = 0$ . Finalement, l'algorithme se décompose en trois phases, une première phase où seule la reconstruction est réalisée, 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 matrice à inverser dans (7) est mal conditionnée, la fenêtre  $W$  peut alors être diminué et par suite la période d'acquisition des images  $\Delta T$  est diminuée permettant ainsi d'augmenter  $\lambda$ .

Enfin, toutes les expérimentations ont été réalisées dans les conditions suivantes :  $\lambda = 0.3$ ,  $Z^* = 65$  cm,  $W$  a été fixée à

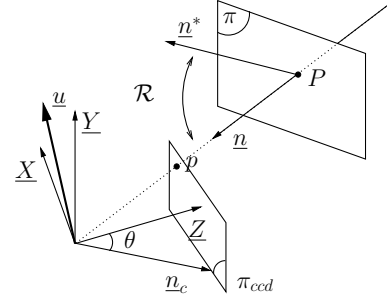


FIG. 1 – Rotation à réaliser par la caméra.

$101 \times 101$  pixels,  $\Delta t$  n'est pas constant, il varie de 120 ms à 200 ms suivant le nombre d'itérations nécessaires à la convergence de (10) (pour la dernière phase les valeurs suivantes ont été utilisées  $\lambda = 1$ ,  $W = 11 \times 11$  pixels,  $\Delta t = 80$  ms).

La première expérimentation concerne la réalisation d'une tâche de positionnement en face de l'objet. La figure 2.a décrit l'évolution de  $\underline{\Theta}$  dans un repère fixe (filtré et non filtré) ; la figure 2.b les composantes de  $\underline{n}_c$  ; la figure 2.c les composantes de  $T_c$  ; la figure 2.d la norme de  $\underline{e}$  ; la figure 2.e les composantes de  $\underline{d}$  ; la figure 2.f la profondeur estimée. Finalement, les images initiale et finale sont rapportées respectivement sur les figures 2.g-h.

Tout d'abord, la figure 2.d confirme que la loi de commande converge puisque  $\|\underline{e}\|$  tend vers zéro, tout comme  $\hat{Z}$  (figure 2.f) vers  $Z^*$ . De même, le point  $p$  tend vers le centre du capteur, pratiquement en ligne droite (voir la figure 2.h où la trajectoire de  $p$  est dessinée, le premier segment de droite correspond au mouvement relatif à la phase initiale). Ce décalage par rapport à une trajectoire linéaire est vraisemblablement dû à un mauvais suivi de  $p$ . En revanche, on peut remarquer sur la figure 2.e que des déplacements importants d'une image à l'autre sont pris en compte. Pour cette expérimentation, l'orientation initiale de la caméra était  $\phi_X = 8.8^\circ$ ,  $\phi_Y = -14.6^\circ$ , l'orientation après asservissement devient  $\phi_X = 1.5^\circ$  et  $\phi_Y = -1.3^\circ$  (rappelons que  $\phi_Z$  n'est pas contrôlé). Par conséquent, le positionnement est relativement précis.

La seconde expérimentation concerne cette fois l'orientation désirée suivante :  $\phi_X = 10^\circ$  et  $\phi_Y = 20^\circ$ . Le comportement de l'algorithme est rapporté sur la figure 3. Les figures 3.a-h 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 mais aussi sur la trajectoire de  $p$ . Ce problème reste néanmoins sans conséquence puisque l'erreur de positionnement est faible. En effet, l'orientation finale mesurée est  $\phi_X = 9.2^\circ$  et  $\phi_Y = 19.8^\circ$ , l'orientation initiale restant inchangée.

## 6 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, de faibles erreurs d'orientation ont pu être observées ( $\approx 2^\circ$ ), elles restent similaires à celles obtenues lors de notre précédente approche [1], mais les vitesses ont pu être augmentées d'un facteur 10. Regrettons cependant que la méthode soit sensible à la calibration du porteur comme à celle de la caméra.

Une perspective intéressante à ce travail concerne 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] A. Alhaj, C. Collewet, et F. Chaumette. Visual servoing based on dynamic vision. Dans *IEEE Int. Conf. on Robotics and Automation, ICRA'2003*, Taipei, Taiwan, 14-19 Septembre, 2003.
- [2] 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.
- [3] A. Crétual et F. Chaumette. Visual servoing based on image motion. *Int. Journal of Robotics Research*, 20(11) :857–877, Novembre 2001.
- [4] F.X. Espiau, E. Malis, et P. Rives. Robust features tracking for robotic applications : towards 2d1/2 visual servoing with natural images. Dans *IEEE Int. Conf. on Robotics and Automation, ICRA'2002*, Washington, USA, 11-15 Mai, 2002.
- [5] G. Hager et K. Toyama. Incremental focus of attention for robust visual tracking. *International Journal of Computer Vision*, 35(1) :45–63, Novembre 1999.
- [6] B.K.P. Horn et B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 16(1–3) :185–203, Août 1981.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] J. Shi et C. Tomasi. Good features to track. Dans *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, pages 593–600, Seattle, Juin 1994.
- [11] B. Yoshimi et P. K. Allen. Active uncalibrated visual servoing. Dans *IEEE Int. Conf. on Robotics and Automation, ICRA'94*, pages 156–161, San Diego, Mai 1994.

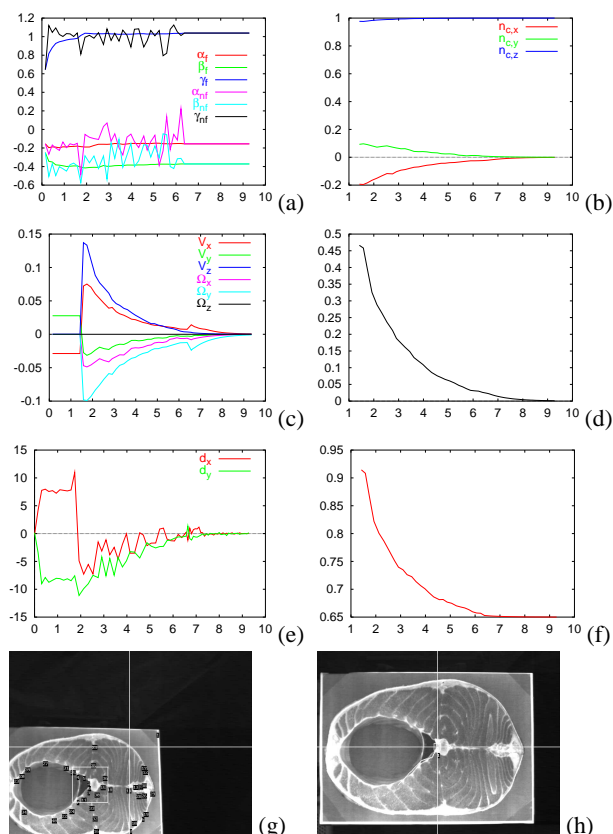


FIG. 2 – 1<sup>re</sup> expérimentation (abscisses en seconde) : (a) Composantes de  $\Theta^f$  (filtrées et non filtrées). (b) Composantes de  $\underline{n}_c$ . (c) Composantes de  $T_c$  (m/s ou rad/s). (d) Erreur définie par  $\|e\|$ . (e) Composantes de  $\underline{d}$  (pixel). (f) Profondeur désirée (m). (g) Image initiale. (h) Image finale.

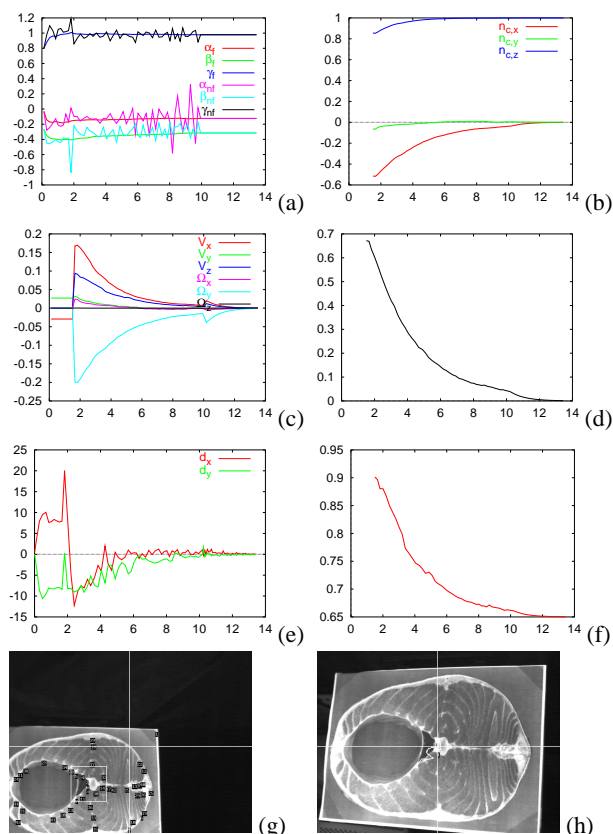


FIG. 3 – 2<sup>e</sup> expérimentation.