

Asservissement visuel basé sur des informations photométriques

Visual servoing based on photometric features

Christophe Collewet

Eric Marchand

François Chaumette

INRIA Rennes-Bretagne Atlantique, IRISA, Lagadic, Rennes, France.

prenom.nom@irisa.fr

Résumé

Cet article propose une nouvelle façon d'aborder la réalisation de tâches robotiques par asservissement visuel. Contrairement aux approches classiques où l'image est utilisée pour extraire des informations de nature géométrique (points, droites, moments, etc.), les informations visuelles utilisées sont directement la luminance en tout point de l'image. Diverses lois de commande proposées dans la littérature étant alors mises en échec, nous reformulons le problème de l'asservissement visuel comme un problème d'optimisation aboutissant à l'écriture d'une nouvelle loi de commande. Des résultats expérimentaux valident l'approche proposée et montrent sa robustesse vis-à-vis de l'approximation faite sur les profondeurs, de scènes non Lambertiennes ou encore d'occultations partielles de la scène.

Mots Clef

Asservissement visuel, luminance, informations visuelles, tâche de positionnement.

Abstract

This paper proposes a new way to achieve robotic tasks by visual servoing. Instead of using geometric features (points, straight lines, moments, etc.) as it is usually done, we use directly the luminance of all pixels in the image. Since most of the classical control laws fail in this case, we turn the visual servoing problem into an optimization problem leading to a new control law. Experimental results validate the proposed approach and show its robustness regarding to approximated depths, non Lambertian objects and partial occlusions.

Keywords

Visual servoing, luminance, visual features, positioning tasks.

1 Introduction

Les asservissements visuels permettent le contrôle des mouvements d'un système dynamique à partir d'informa-

tions visuelles acquises par une ou plusieurs caméras embarquées ou déportées [7]. Ce type d'approches nécessite en général une extraction et un suivi robuste de ces informations visuelles, ce qui peut parfois s'avérer problématique. Dans cet article, nous montrons que nous pouvons nous affranchir de tout traitement des images. En effet, nous nous proposons d'utiliser directement l'intensité de l'image dans la loi de commande.

La loi de commande permet le déplacement d'un robot à une position souhaitée à partir de la connaissance des informations visuelles courantes s et désirées s^* . Son principe repose sur la régulation à zéro du vecteur d'erreur $s - s^*$. Pour ce faire, cette approche nécessite la connaissance d'une matrice dite d'interaction [4] exprimant la relation liant la vitesse de la caméra v par rapport à l'objet observé à la vitesse des informations visuelles choisies

$$\dot{s} = L_s v \quad (1)$$

où la vitesse v peut être décomposée en une composante de translation $v = (v_x, v_y, v_z)$ et de rotation $\omega = (\omega_x, \omega_y, \omega_z)$. Dans ce type d'approche la matrice d'interaction joue un rôle important, en effet, en supposant que l'entrée du contrôleur soit directement la vitesse v , une loi de commande tentant d'assurer au mieux une décroissance exponentielle découplée de l'erreur $s - s^*$ est donnée par

$$v = -\lambda \widehat{L}_s^+ (s - s^*) \quad (2)$$

où le gain λ permet de régler la vitesse de convergence et où \widehat{L}_s^+ est la pseudo-inverse d'un modèle ou d'une approximation de L_s [4].

Un des points clés de cette approche concerne le choix des informations visuelles. Ces informations visuelles, issues de mesures $x(r_k)$ dans l'image (où r_k est la pose de la caméra à l'instant k) peuvent être extrêmement variées. Elles peuvent être tout aussi bien de type 2D (coordonnées de points dans l'image, moments, ...) que de type 3D en combinant les mesures 2D issues du capteur avec des connaissances supplémentaires (paramètres intrinsèques de la caméra, modèle 3D de l'objet par exemple). Dans tous les cas, une extraction robuste, une mise en correspondance

(entre $\mathbf{x}(\mathbf{r}_k)$ et $\mathbf{x}^* = \mathbf{x}(\mathbf{r}^*)$) et un suivi spatio-temporel (entre $\mathbf{x}(\mathbf{r}_{k-1})$ et $\mathbf{x}(\mathbf{r}_k)$) doivent être réalisés. Ces différentes phases sont réputées pour être difficiles, comme l’atteste l’abondante littérature sur ces sujets [12].

Nous nous proposons dans cet article de modifier radicalement cette façon de faire en supprimant la phase d’extraction des mesures, et par suite en supprimant aussi les phases de mise en correspondances et de suivi. Pour ce faire, nous utilisons l’information visuelle la plus simple qui soit : l’intensité lumineuse. Dans ce cas, le vecteur d’informations visuelles \mathbf{s} n’est rien d’autre que l’image, \mathbf{s}^* étant simplement l’image désirée. Dans ces conditions, le signal d’erreur $\mathbf{s} - \mathbf{s}^*$ est la différence entre les images courante et désirée (c’est-à-dire $\mathbf{I} - \mathbf{I}^*$ où \mathbf{I} est un vecteur contenant l’intensité de tous les pixels de l’image). Dans ce premier travail, nous supposons que la scène observée est Lambertienne, ce qui nous permet d’obtenir la forme analytique de la matrice d’interaction relative à la luminance. Elle peut être obtenue [11] à partir de l’hypothèse d’invariance temporelle de la luminance [6].

Toutefois, en utilisant pour information visuelle l’intensité de l’image, la loi de commande classiquement utilisée (voir l’équation (2), au mieux converge, mais en générant des mouvements 3D inappropriés de la caméra, ou au pire diverge. Nous reformulons alors le problème de l’asservissement visuel comme celui de la minimisation d’une fonction de coût à l’instar des travaux rapportés dans [10] ou [13]. Nous montrons ensuite comment il est possible de réaliser des tâches de positionnement à partir d’une nouvelle commande.

Considérer toute l’image comme une information visuelle a déjà été proposé dans le passé [2, 3, 14]. En effet, comme dans notre cas, ces approches ne nécessitent aucunement une phase de mise en correspondance. Néanmoins, elles diffèrent de notre approche selon deux points importants. Tout d’abord, elles n’utilisent pas directement l’intensité de l’image puisqu’une projection dans un espace propre est utilisée pour réduire la dimension du vecteur \mathbf{s} . La loi de commande opère alors directement à partir de l’espace propre et non plus directement sur les intensités. Cela nécessite d’une part le calcul de cet espace propre (calcul réalisé hors ligne) mais d’autre part aussi la projection dans cet espace de toute nouvelle image. Par ailleurs, la matrice d’interaction utilisée est obtenue, non pas de façon analytique, mais par apprentissage (également hors ligne). Cette phase d’apprentissage présente deux inconvénients majeurs : l’apprentissage doit être réalisé pour tout nouvel objet et il requiert un grand nombre d’images de la scène pour diverses positions de la caméra. Ces inconvénients sont évités ici grâce à la forme analytique de la matrice d’interaction.

Une approche intéressante, considérant également les intensités, a été proposée dans [8]. Elle est basée sur des méthodes de noyau permettant d’atteindre un haut degré de découplage de la loi de commande. Toutefois, seules les translations et la rotation suivant l’axe optique ont été

considérées alors que notre approche considère les 6 degrés de liberté.

Dans [15], un asservissement visuel basé homographie est présenté. Dans ce travail, les intensités de l’image sont utilisées pour estimer une homographie entre les images courante et désirée par calcul d’une mise en correspondance, homographie qui est ensuite utilisée dans la loi de commande. Bien que, comme dans notre cas, l’intensité de l’image soit à la base de cette approche, un effort important de traitement d’images doit être réalisé pour obtenir cette homographie.

Cet article est structuré comme suit. Le problème de l’asservissement visuel est reformulé dans la section 2 en terme d’un problème d’optimisation. Le calcul de la matrice d’interaction associée à la luminance est rappelé dans la section 3. L’allure de la fonction à minimiser est étudiée quant à elle dans la section 4. La section 5 détaille comment il est possible de réaliser des tâches de positionnement. Les résultats expérimentaux sont présentés dans la section 6.

2 Formulation sous la forme d’un problème d’optimisation

Au lieu de formuler le problème de l’asservissement visuel de manière classique, comme évoqué dans la section 1, il est possible, comme dans [10] ou [13], de le reformuler en cherchant à minimiser la fonction de coût suivante

$$\mathcal{C}(\mathbf{r}) = (\mathbf{s}(\mathbf{r}) - \mathbf{s}(\mathbf{r}^*))^\top (\mathbf{s}(\mathbf{r}) - \mathbf{s}(\mathbf{r}^*)) \quad (3)$$

où \mathbf{r} décrit la pose courante de la caméra par rapport à l’objet, c’est-à-dire un élément de $\mathbb{R}^3 \times SO(3)$, et \mathbf{r}^* la pose désirée. Nous noterons « + » l’opérateur consistant à combiner deux changements de repère.

Suivant la méthode de minimisation employée, diverses lois de commande peuvent être exhibées. Elles sont détaillées dans [10], aussi nous ne rapportons ci-après que les résultats les plus importants, en nous restreignant, nous aussi, aux schémas de convergence différentiels plus à même de se transposer en terme d’une loi de commande. Ils sont du type

$$\mathbf{r} = \mathbf{r}_k + t_k \mathbf{d}(\mathbf{r}_k) \quad (4)$$

où \mathbf{r}_k est la pose courante, t_k un scalaire positif (le pas de descente) et $\mathbf{d}(\mathbf{r}_k)$ une direction de descente assurant la décroissance de (3) si la condition suivante est respectée

$$\mathbf{d}(\mathbf{r}_k)^\top \nabla \mathcal{C}(\mathbf{r}_k) < 0. \quad (5)$$

Dans ce cas, si t_k est suffisamment faible, on en déduit la loi de commande suivante

$$\mathbf{v} = \lambda_k \mathbf{d}(\mathbf{r}_k) \quad (6)$$

où λ_k est un scalaire fonction de t_k et de la période d’échantillonnage choisie. Généralement λ_k est choisi constant pour tout k .

Pour ne pas alourdir les notations et en l’absence d’ambiguïté, nous omettrons désormais les indices k .

2.1 Méthode du gradient

La direction de descente est alors simplement

$$\mathbf{d}(\mathbf{r}) = -\nabla\mathcal{C}(\mathbf{r}) \quad (7)$$

où

$$\nabla\mathcal{C}(\mathbf{r}) = \left(\frac{\partial\mathbf{s}}{\partial\mathbf{r}}\right)^\top (\mathbf{s}(\mathbf{r}) - \mathbf{s}(\mathbf{r}^*)). \quad (8)$$

Or, puisque nous avons $\dot{\mathbf{s}} = \frac{\partial\mathbf{s}}{\partial\mathbf{r}}\dot{\mathbf{r}} = \mathbf{L}_s\mathbf{v}$, on obtient la loi de commande

$$\mathbf{v} = -\lambda\mathbf{L}_s^\top (\mathbf{s}(\mathbf{r}) - \mathbf{s}(\mathbf{r}^*)). \quad (9)$$

Cette méthode a été appliquée par exemple dans [5].

2.2 Méthode de Gauss-Newton

Pour \mathbf{r}_k dans un voisinage de \mathbf{r}^* , en linéarisant $\mathbf{s}(\mathbf{r})$ autour de $\mathbf{s}(\mathbf{r}_k)$ puis en reportant cette expression linéarisée dans (3) et en annulant ensuite le gradient de l'expression obtenue, on obtient

$$\mathbf{d}(\mathbf{r}) = -(\mathbf{L}_s^\top\mathbf{L}_s)^{-1}\nabla\mathcal{C}(\mathbf{r}) \quad (10)$$

soit en utilisant (8)

$$\mathbf{v} = -\lambda\mathbf{L}_s^+ (\mathbf{s}(\mathbf{r}) - \mathbf{s}(\mathbf{r}^*)) \quad (11)$$

qui n'est autre que la version parfaite de (2), c'est-à-dire quand tous les termes de \mathbf{L}_s sont parfaitement connus. Dans le cas contraire, on utilise

$$\mathbf{v} = -\lambda\mathbf{L}_{s^*}^+ (\mathbf{s}(\mathbf{r}) - \mathbf{s}(\mathbf{r}^*)). \quad (12)$$

2.3 Méthode de Newton

En approximant localement $\mathcal{C}(\mathbf{r})$ par son développement en série de Taylor en \mathbf{r} , et en annulant son gradient, il vient

$$\mathbf{d}(\mathbf{r}) = -(\nabla^2\mathcal{C}(\mathbf{r}))^{-1}\nabla\mathcal{C}(\mathbf{r}) \quad (13)$$

avec

$$\nabla^2\mathcal{C}(\mathbf{r}) = \mathbf{L}_s^\top\mathbf{L}_s + \sum_{i=1}^{i=\dim\mathbf{s}} \nabla^2 s_i (s_i(\mathbf{r}) - s_i(\mathbf{r}^*)). \quad (14)$$

Remarquons que le vecteur \mathbf{d} sera effectivement une direction de descente si $\nabla^2\mathcal{C}(\mathbf{r}) > 0$. Remarquons également qu'en \mathbf{r}^* les méthodes de Newton et de Gauss-Newton sont équivalentes puisque $\nabla^2\mathcal{C}(\mathbf{r}^*) = \mathbf{L}_{s^*}^\top\mathbf{L}_{s^*}$.

Cette approche a montré son efficacité dans [9].

2.4 Méthode de Levenberg-Marquardt

Cette méthode considère la direction de descente suivante

$$\mathbf{d}(\mathbf{r}) = -(\mathbf{G} + \mu \text{diag}(\mathbf{G}))^{-1}\nabla\mathcal{C}(\mathbf{r}) \quad (15)$$

où l'on choisit habituellement $\mathbf{G} = \nabla^2\mathcal{C}(\mathbf{r})$ ou plus simplement

$\mathbf{G} = \left(\frac{\partial\mathbf{s}}{\partial\mathbf{r}}\right)^\top \left(\frac{\partial\mathbf{s}}{\partial\mathbf{r}}\right)$ conduisant dans ce cas à

$$\mathbf{v} = -\lambda(\mathbf{H} + \mu \text{diag}(\mathbf{H}))^{-1}\mathbf{L}_s^\top (\mathbf{s}(\mathbf{r}) - \mathbf{s}(\mathbf{r}^*)) \quad (16)$$

avec $\mathbf{H} = \mathbf{L}_s^\top\mathbf{L}_s$. Le paramètre μ permet de passer de la méthode du gradient à celle de Gauss-Newton en observant le comportement de la fonction de coût pendant sa minimisation. En effet, quand μ est élevé, (16) se comporte comme (9)¹. En revanche, quand μ est faible, (16) se comporte comme (11).

2.5 Méthode ESM

Les auteurs de [10] ont proposé une loi de commande se comportant comme une méthode du second ordre sans toutefois nécessiter le calcul de $\nabla^2\mathcal{C}(\mathbf{r})$. Elle s'écrit

$$\mathbf{v} = -2\lambda(\mathbf{L}_s + \mathbf{L}_{s^*})^+ (\mathbf{s}(\mathbf{r}) - \mathbf{s}(\mathbf{r}^*)). \quad (17)$$

Par rapport aux algorithmes classiques de minimisation, cette approche tire profit du fait que l'allure de la fonction de coût est connue au minimum, ce qui n'est que rarement le cas dans d'autres problèmes de minimisation.

3 Modélisation de la luminance

Les informations visuelles considérées dans cet article sont la luminance en tous les points de l'image, c'est-à-dire,

$$\mathbf{s}(\mathbf{r}) = \mathbf{I}(\mathbf{r}) = (\mathbf{I}_{1\bullet}, \mathbf{I}_{2\bullet}, \dots, \mathbf{I}_{N\bullet}) \quad (18)$$

où $\mathbf{I}_{k\bullet}$ n'est autre que la ligne k de l'image. $\mathbf{I}(\mathbf{r})$ est donc un vecteur de taille $N \times M$ où $N \times M$ est la taille de l'image.

Comme nous l'avons constaté dans la Section 2, toutes les lois de commande nécessitent le calcul de la matrice d'interaction, ou pour le moins une approximation de cette matrice. Dans notre cas, il s'agit de celle associée à la luminance d'un point de l'image. Ce calcul est détaillé dans [11], il est toutefois rappelé ci-après².

L'hypothèse de base de ce calcul repose sur l'invariance de la luminance pour un même point physique entre deux instants. Cette hypothèse permet d'obtenir l'équation de contrainte du mouvement apparent (ECMA) qui lie la variation temporelle de la luminance I au mouvement 2D du point de coordonnées \mathbf{x} dans l'image.

Plus précisément, en supposant que ce point se déplace de $d\mathbf{x}$ dans l'intervalle de temps dt , l'hypothèse précédente permet d'écrire [6] :

$$I(\mathbf{x} + d\mathbf{x}, t + dt) = I(\mathbf{x}, t). \quad (19)$$

Un développement en série de Taylor au premier ordre en \mathbf{x} suivi d'une division par dt conduit ensuite à

$$\nabla I^\top \dot{\mathbf{x}} + \frac{\partial I}{\partial t} = 0. \quad (20)$$

¹Plus précisément, chaque composante du gradient est mis à l'échelle suivant la diagonale de la matrice Hessienne, cela permet de plus grands déplacements dans la direction où le gradient est faible.

²Dans [11], cette matrice d'interaction a été utilisée pour contrôler la position d'une source lumineuse et de la caméra pour réaliser un éclairage optimal de la scène.

Il est alors possible d'obtenir la matrice d'interaction recherchée

$$\frac{\partial I}{\partial \mathbf{t}} = \mathbf{L}_I \mathbf{v} \quad (21)$$

en introduisant dans (20) la matrice d'interaction \mathbf{L}_x associée à celle du point \mathbf{x} [4], il vient alors

$$\mathbf{L}_I = -\nabla \mathbf{I}^\top \mathbf{L}_x \quad (22)$$

où

$$\mathbf{L}_x = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \quad (23)$$

Bien entendu, en considérant les hypothèses nécessaires à la dérivation de l'ECMA, l'équation (22) n'est valable que dans le cas d'un environnement Lambertien, c'est-à-dire diffusant la lumière de façon uniforme dans toutes les directions, mais aussi pour une source de lumière fixe par rapport à l'objet. Ces hypothèses peuvent paraître restrictives, toutefois, nous verrons à la section 6 que l'utilisation de (22) s'avère être très efficace même lorsque l'hypothèse d'une scène Lambertienne n'est plus valide.

4 Étude de la fonction de coût

La convergence des lois de commande rappelées dans la section 2 dépend fortement de l'allure de la fonction de coût à minimiser, aussi allons-nous nous y intéresser.

Pour ce faire, nous considérons maintenant le vecteur $\mathbf{I}(\mathbf{r})$ donné par (18) lorsqu'il est observé suivant la pose \mathbf{r} . Nous notons \mathbf{r} sous la forme $\mathbf{r} = (\mathbf{t}, \boldsymbol{\theta})$ où $\mathbf{t} = (t_x, t_y, t_z)$ caractérise la partie translation de la matrice homogène décrivant le passage du repère courant de la caméra au repère désiré et où $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$ caractérise sa partie rotation (tangage, lacet, roulis).

À titre d'exemple, les figures 1b et 1c rapportent, dans le sous espace (t_x, θ_y) , l'allure de la fonction de coût (3) quand l'objet (plan) observé est celui représenté sur la figure 1a et lorsque la pose souhaitée est telle que le plan du capteur et celui de l'objet soient parallèles à une distance $Z^* = 80$ cm. Signalons dès à présent que ce cas de figure est en fait le plus complexe (avec son pendant (t_y, θ_x)). En effet, il est bien connu qu'il est très difficile de différencier dans une image un mouvement de translation d'axe x (respectivement y) d'un mouvement de rotation autour de l'axe y (respectivement x). Cela explique la présence d'une direction privilégiée selon laquelle le coût est faible. Cette direction est clairement mise en évidence sur les figures 1b et 1c.

D'une façon générale, des simulations ont montré que la forme de la fonction de coût ne dépend pas trop de l'objet observé tant que l'image ne présente pas de motif périodique ou de forte variation des composantes du gradient spatial de l'image. Elle présente toujours une vallée très abrupte au milieu d'un plateau en pente douce et de pente non constante. Précisons que la direction de cette vallée dépend de Z^* . Pour $Z^* \approx 0$ la direction est celle de l'axe θ_y

et devient celle de l'axe t_x pour $Z^* \rightarrow +\infty$. Remarquons également sur la figure 1b que (3) n'est que quasi-convexe et de plus sur un domaine de l'espace très réduit.

Examinons à présent plus précisément (3) dans un voisinage de \mathbf{r}^* . Pour ce faire, réalisons un développement en série de Taylor au premier ordre des informations visuelles $\mathbf{I}(\mathbf{r})$ en \mathbf{r}^*

$$\mathbf{I}(\mathbf{r}) = \mathbf{I}(\mathbf{r}^*) + \mathbf{L}_{I^*} \Delta \mathbf{r} \quad (24)$$

où $\Delta \mathbf{r}$ correspond à la pose relative pour passer de \mathbf{r} à \mathbf{r}^* . Par suite, en portant (24) dans (3), il vient une approximation du coût dans un voisinage de \mathbf{r}^*

$$\widehat{\mathcal{C}}(\mathbf{r}) = \Delta \mathbf{r}^\top \mathbf{H}^* \Delta \mathbf{r} \quad (25)$$

avec $\mathbf{H}^* = \mathbf{L}_{I^*}^\top \mathbf{L}_{I^*}$. En pratique, due à la forme particulière de la matrice d'interaction donnée en (22) (la présence d'un terme de profondeur dans la partie translation due à (23)), les valeurs propres de la matrice \mathbf{H}^* sont extrêmement différentes³. Malheureusement, vu la complexité de cette matrice, elles ne peuvent être obtenues que de façon numérique. Par suite, si on s'intéresse au sous-espace (t_x, θ_y) (respectivement (t_y, θ_x)) la fonction de coût est un paraboloïde elliptique dont la valeur du grand axe est très grande vis-à-vis de celle du petit axe, conduisant ainsi à des iso-contours quasiment parallèles entre eux comme le montre la figure 1c. De plus, le calcul des vecteurs propres de \mathbf{H}^* permet d'exhiber des directions selon lesquelles le coût décroît soit de façon très faible lorsque la valeur propre associée est très faible, ou inversement de façon importante lorsque la valeur propre est élevée. Dans le cas de la figure 1c, le vecteur propre associé à la plus petite valeur propre correspond à la vallée dans laquelle le coût varie très lentement. En revanche, il varie localement fortement dans une direction orthogonale, c'est-à-dire dans une direction très voisine de $\nabla \mathcal{C}(\mathbf{r})$. Nous utiliserons cette connaissance du comportement de la la fonction de coût dans la section suivante.

5 Tâches de positionnement

Comme nous l'avons vu dans la section 2, diverses lois de commande permettent de minimiser la fonction de coût (3). Nous rapportons tout d'abord le comportement des lois de commande les plus souvent utilisées dans le domaine de l'asservissement visuel. C'est-à-dire, la méthode de Gauss-Newton (GN) et la méthode ESM (ESM) qui s'est révélée être très efficace pour résoudre, en particulier, le problème dit de l'avance et du retrait [1].

Trois tâches de positionnement ont été simulées, les conditions initiales étant pour la première tâche $t_x = -2$ cm, $\theta_y = 0$, pour la seconde $t_x = 4$ cm, $\theta_y = -3^\circ$ et pour la troisième $t_x = 0$, $\theta_y = -3^\circ$. Les chemins dans le sous-espace (t_x, θ_y) empruntés pour les différentes tâches et pour les

³Ce résultat est également valide pour la plupart des informations visuelles géométriques où un terme de profondeur intervient aussi dans la partie translation.

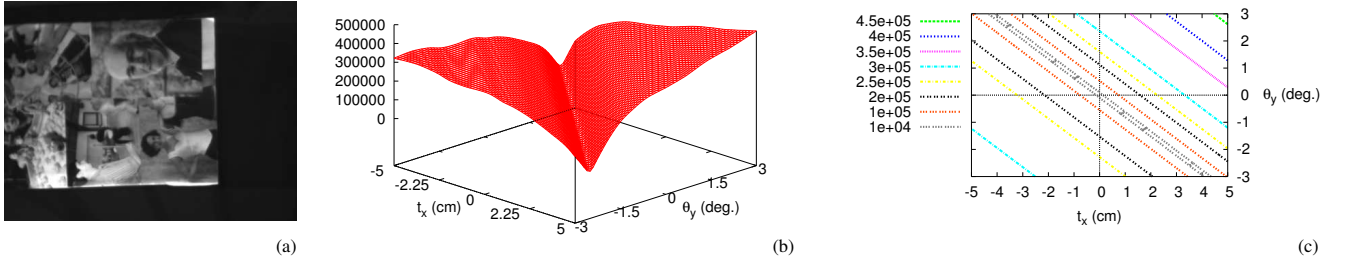


FIG. 1 – Fonction de coût : (a) Objet observé ; (b) Allure de la fonction de coût dans le sous-espace (t_x, θ_y) ; (c) Iso-contours dans le sous-espace (t_x, θ_y) .

différentes lois de commande sont représentés sur la figure 2 (rappelons toutefois que tous les axes de la caméra sont contrôlés).

Dans tous les cas la méthode de Gauss-Newton conduit à des déplacements 3D très grands en regard des déplacements initiaux effectués par la caméra. La méthode ESM a un meilleur comportement. Néanmoins, les chemins empruntés ne sont pas non plus optimaux.

D'une façon générale, les différentes lois de commande ont un comportement relativement similaire, la direction suivie est tout d'abord quasiment parallèle à la vallée (voir les figures 2 et 1c), elle n'est modifiée qu'une fois la fin de celle-ci rencontrée où là, la minimisation se fait alors dans la direction du minimum sans problème. Dans cette dernière phase, les méthodes GN et ESM ont un comportement identique. C'est le cas de figure rencontré dans la deuxième tâche où la position initiale de la caméra était déjà dans la vallée. Comme on peut le constater les deux approches sont équivalentes. De plus, dans ce cas, la convergence est toujours assurée.

À la vue de la fonction de coût obtenue, nous proposons l'algorithme suivant permettant de converger vers le minimum global. La caméra se déplace tout d'abord de telle sorte à atteindre la vallée, puis se déplace ensuite dans cette vallée pour atteindre le minimum. Le premier mouvement peut-être facilement réalisé en utilisant une approche de type gradient comme nous l'avons vu à la fin de la section 4. Toutefois, comme nous pouvons le constater sur la figure 1c, la direction de $\nabla \mathcal{C}(\mathbf{r})$ est constante mais, sur le plateau, son amplitude ne l'est pas (voir la figure 1b), la pente étant variable. Il conviendrait donc de faire varier le paramètre λ impliqué dans (9) pour garantir des vitesses relativement douces. Toutefois, une approche plus simple est possible. Elle consiste à appliquer la loi de commande suivante

$$\mathbf{v} = -v_c \frac{\nabla \mathcal{C}(\mathbf{r}_{\text{init}})}{\|\nabla \mathcal{C}(\mathbf{r}_{\text{init}})\|}. \quad (26)$$

C'est-à-dire à appliquer une vitesse constante de norme v_c dans la direction de la plus grande pente calculée à la position initiale. Le passage dans la vallée se détecte par l'application d'un filtre polynômial sur les différentes valeurs passées du coût, enregistrées au cours du mouvement. Cette détection pouvant être réalisée de manière grossière,

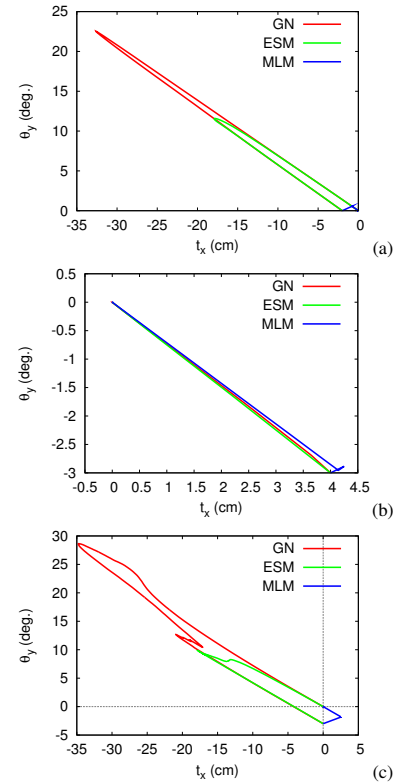


FIG. 2 – Chemin dans le sous-espace (t_x, θ_y) des approches GN, ESM et MLM pour différentes tâches de positionnement. (a) tâche 1 ; (b) tâche 2 ; (c) tâche 3.

la loi de commande (16) est ensuite appliquée pour affiner le positionnement. Toutefois, au lieu de régler le paramètre μ comme dans l'algorithme de Levenberg-Marquardt, nous proposons une autre façon de le régler comme nous le verrons par la suite. Nous nommerons MLM cette méthode dans le reste du document. De plus, au lieu d'utiliser pour la matrice \mathbf{H} la Hessienne de la fonction de coût, nous utiliserons son approximation $\mathbf{L}_I^\top \mathbf{L}_I$. La loi de commande résultante est donc donnée par

$$\mathbf{v} = -\lambda (\mathbf{H} + \mu \text{diag}(\mathbf{H}))^{-1} \mathbf{L}_I^\top (\mathbf{I}(\mathbf{r}) - \mathbf{I}(\mathbf{r}^*)) \quad (27)$$

avec $\mathbf{H} = \mathbf{L}_I^\top \mathbf{L}_I$.

Nous détaillons maintenant comment le paramètre μ est réglé. La Figure 3a montre les chemins obtenus par l’algorithme MLM dans le cas où $\mathbf{r}_{\text{init}} = (8 \text{ cm}, 4 \text{ cm}, -10 \text{ cm}, 3^\circ, -3^\circ, -5^\circ)$ pour différentes valeurs de μ . Après la phase à mouvement constant (donné par (26)), lorsqu’une forte valeur de μ est utilisée, le fond de la vallée est aisément atteint (voir la figure 3a pour $\mu = 1$) puisque (27) se comporte comme une approche de type gradient. Toutefois, puisque cette vallée est très étroite, le déplacement est tel qu’il rebondit sur les bords de la vallée et le taux de convergence vers le minimum (dont la direction est l’axe de la vallée) est extrêmement faible (voir la figure 3b)⁴. En revanche, si μ est trop faible, la loi de commande (27) se comporte comme une approche de type Gauss-Newton (GN) et la convergence n’est plus assurée (voir le grand déplacement observé sur la figure 3a quand $\mu = 10^{-3}$). Une valeur intermédiaire ($\mu = 10^{-2}$) doit donc être utilisée pour assurer à la fois un chemin optimal (Fig. 3a) et un bon taux de convergence (voir la figure 3b). C’est cette valeur qui sera utilisée dans la suite de ce document aussi bien pour les simulations que pour les expérimentations.

Le comportement de la méthode MLM pour les tâches de positionnement effectuées précédemment par les méthodes GN et ESM est détaillé sur la figure 2. Elle montre clairement que les chemins obtenus sont cette fois beaucoup plus optimisés.

6 Résultats expérimentaux

Les expérimentations ont été réalisées sur la plate-forme expérimentale de l’IRISA. Elle est composée d’une caméra CCD montée sur l’effecteur d’un robot AFMA à six degrés de liberté. Signalons que certaines vidéos des expérimentations détaillées ci-dessous sont accessibles à partir de la page <http://www.irisa.fr/lagadic/demo-visual-servoing.html>.

La première expérimentation concerne un objet plan (une nappe) pouvant être assimilé à un objet Lambertien de telle sorte que la matrice donnée par (22) puisse être considérée comme valide. La pose initiale était $\mathbf{r}_{\text{init}} = (20 \text{ cm}, 0 \text{ cm}, 0 \text{ cm}, 0^\circ, 11^\circ, 0^\circ)$, c’est-à-dire très proche de la vallée. La pose désirée était telle que le plan de l’objet et le plan image étaient parallèles. La matrice d’interaction a été calculée à chaque itération en supposant toutes les profondeurs constantes et égales à $Z^* = 80 \text{ cm}$, ce qui est une approximation grossière. La figure 4a rapporte le comportement de la fonction de coût lorsque la méthode GN a été utilisée, la figure 4b celui observé quand la méthode MLM a été utilisée. Les images initiales et finales sont représentées respectivement sur les figures 4c et 4d ; les figures 4e et 4f décrivent respectivement le signal d’erreur $\mathbf{I} - \mathbf{I}^*$ pour les positions initiale et finale. Tout d’abord, comme le montre les figures 4a et b, les deux lois de commande convergent puisque les fonctions de coût s’annulent. Toutefois, la vitesse de convergence de la méthode GN est

⁴Augmenter la valeur de λ n’augmente pas le taux de convergence mais l’amplitude des oscillations de part et d’autre de la vallée.

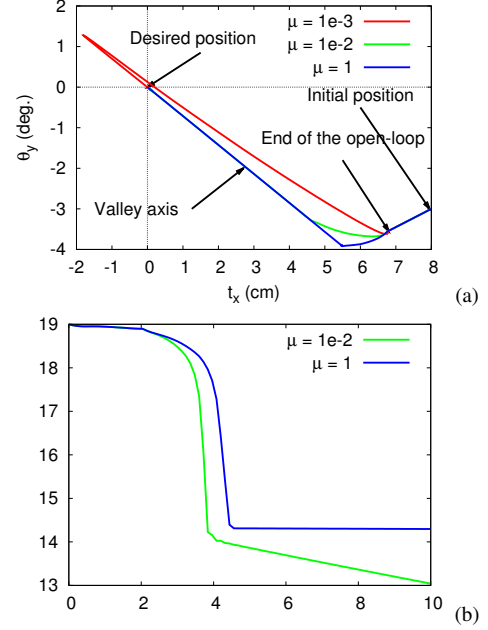


FIG. 3 – Influence de μ . (a) Chemin dans le sous-espace (t_x, θ_y) pour $\mathbf{r} = (8 \text{ cm}, 4 \text{ cm}, -10 \text{ cm}, 3^\circ, -3^\circ, -5^\circ)$, (b) Logarithme de la fonction de coût en fonction du temps (en s).

très faible comparée à l’approche MLM. Remarquons également que le réglage du paramètre λ impliqué dans (11) n’est pas aisé pour la méthode GN, un compromis doit être réalisé entre oscillations à la fin du mouvement et des vitesses acceptables au tout début du mouvement. Il a donc été fixé à 5 au début du mouvement puis à 1 à la fin. Pour la méthode MLM une valeur constante $\lambda = 4$ a été utilisée. Par ailleurs, nous pouvons constater sur la figure 4b la première phase de l’algorithme où un mouvement constant est utilisé (vers $t \approx 2.5 \text{ s}$). En effet, le coût décroît puis recroît afin que le passage dans la vallée puisse être détecté par le filtre.

La seconde expérimentation est beaucoup plus complexe que la première. Tout d’abord la pose désirée n’est plus telle que les deux plans soient parallèles, une rotation d’axe x de 5° a été réalisée. Par conséquent, les profondeurs désirées sont maintenant toutes différentes. Néanmoins, nous avons conservé la valeur $Z = 80 \text{ cm}$ dans la matrice d’interaction pour montrer ainsi la robustesse de la loi de commande aux variations de profondeur. De plus, un objet non Lambertien a été utilisé, il s’agit d’une photographie recouverte d’une plaque de verre. Comme le montre la figure 5 où cet objet est représenté, l’objet utilisé est très spéculaire. La pose initiale est également plus compliquée dans le sens où elle conduit à un plus grand déplacement pour atteindre la position désirée. En effet, nous avons maintenant $\mathbf{r}_{\text{init}} = (20 \text{ cm}, 10 \text{ cm}, 5 \text{ cm}, 10^\circ, 11^\circ, 15^\circ)$. Le comportement de la méthode MLM est décrit sur la figure 6. Plus précisément, la figure 6a rapporte la vitesse de la caméra ;

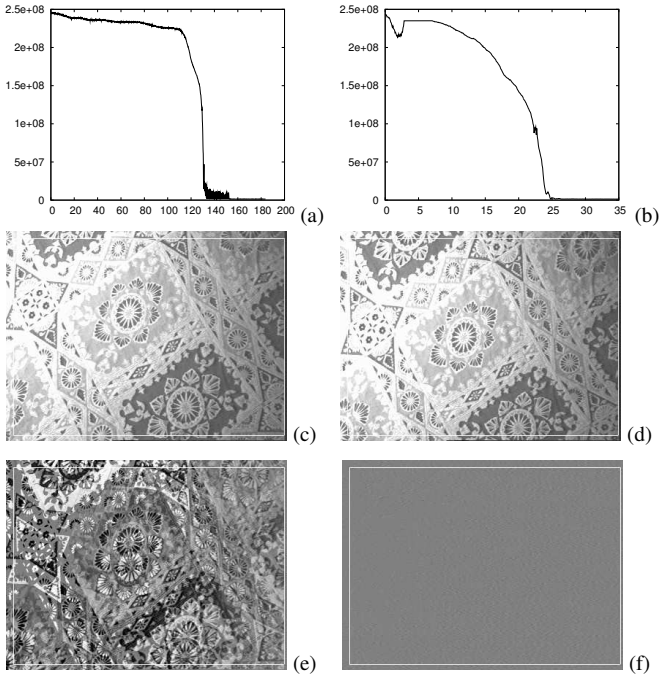


FIG. 4 – Première expérimentation. Objet Lambertien (axe des x en seconde). (a) Fonction de coût en utilisant la méthode GN, (b) Fonction de coût en utilisant la méthode MLM, (c) Image initiale, (d) Image finale, (e) $I - I^*$ en position initiale, (f) $I - I^*$ en position finale.



FIG. 5 – L'objet non Lambertien utilisé.

la figure 6b le comportement de la fonction de coût ; la figure 6c la partie translation de la pose $\Delta \mathbf{r}$ entre \mathbf{r} et \mathbf{r}^* (les rotations sont représentées à l'aide d'un vecteur unitaire porté par l'axe de la rotation et par un angle autour de cet axe). Comme nous le constatons, malgré l'utilisation d'un objet non Lambertien et les différentes approximations utilisées, la loi de commande converge sans problème. Toutefois, la fonction de coût n'étant pas lisse, la vitesse de la caméra est bruitée. Néanmoins, cela n'a qu'un très faible effet sur la trajectoire de la caméra comme le montrent les figures 6c et 6d. Finalement, l'erreur de positionnement est très faible, en effet nous avons $\Delta \mathbf{r} = (-0.1 \text{ mm}, -0.1 \text{ mm}, -0.1 \text{ mm}, -0.01^\circ, -0.01^\circ, -0.01^\circ)$. Cette très faible erreur de positionnement vient du fait que $I - I^*$ est très sensible à la pose \mathbf{r} .

L'expérimentation suivante est focalisée sur les occultations partielles. Les poses initiale et désirée sont inchan-

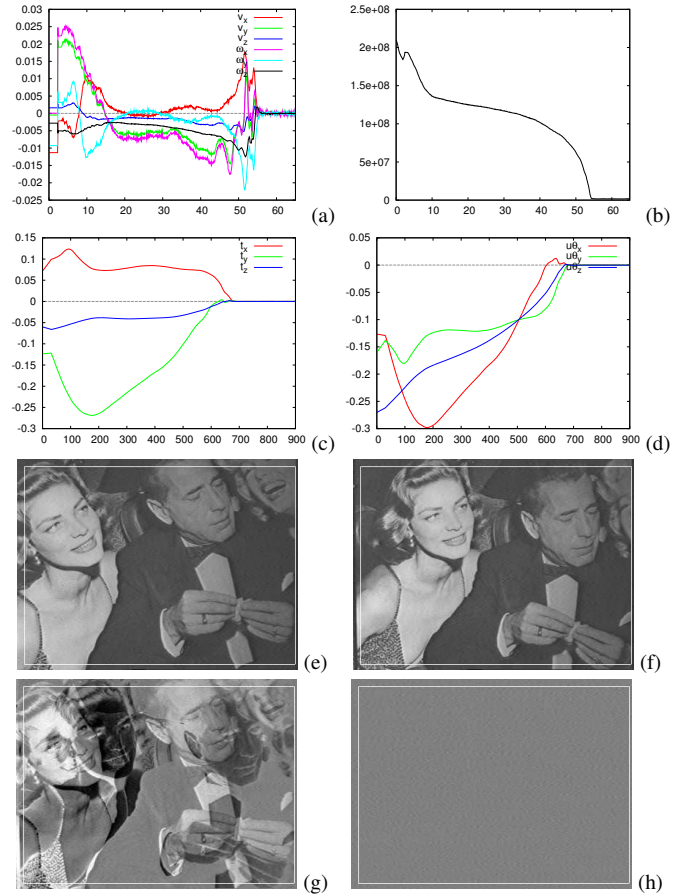


FIG. 6 – Deuxième expérimentation. Méthode MLM (axes des x en seconde pour (a) et (b), en itération pour (c) et (d)). (a) Vitesse de la caméra (m/s ou rad/s), (b) Fonction de coût, (c) Partie translation de $\Delta \mathbf{r}$ (en mètre), (d) Partie rotation de $\Delta \mathbf{r}$ (en radian), (e) Image initiale, (f) Image finale (et désirée), (g) $I - I^*$ pour la position initiale, (h) $I - I^*$ pour la position finale.

gées par rapport à l'expérimentation précédente. Cependant, après avoir amené la caméra à sa position initiale, un objet a été rajouté dans la scène de telle façon que l'image initiale est celle représentée sur la figure 7a alors que l'image désirée reste celle représentée sur la figure 6f. Cet objet est également bougé à la main pendant la réalisation de la tâche comme le montrent les figures 7b et 7c ce qui augmente encore la surface de l'occultation. Malgré cela, la loi de commande converge (voir Fig. 7f). Bien évidemment, en raison de l'occultation, la fonction de coût ne peut plus s'annuler à la fin du mouvement (voir les figures 7f et 7h). Néanmoins, l'erreur de positionnement n'est pas affectée par cette occultation, en effet cette erreur est $\Delta \mathbf{r} = (-0.1 \text{ mm}, -0.2 \text{ mm}, -0.1 \text{ mm}, -0.02^\circ, 0.02^\circ, -0.01^\circ)$ ce qui est très similaire à ce qui a été obtenu lors de l'expérimentation précédente. Ce bon résultat est dû à la forte redondance des informations visuelles utilisées.

Le but de la dernière expérimentation est de montrer la robustesse de la loi de commande aux erreurs commises sur

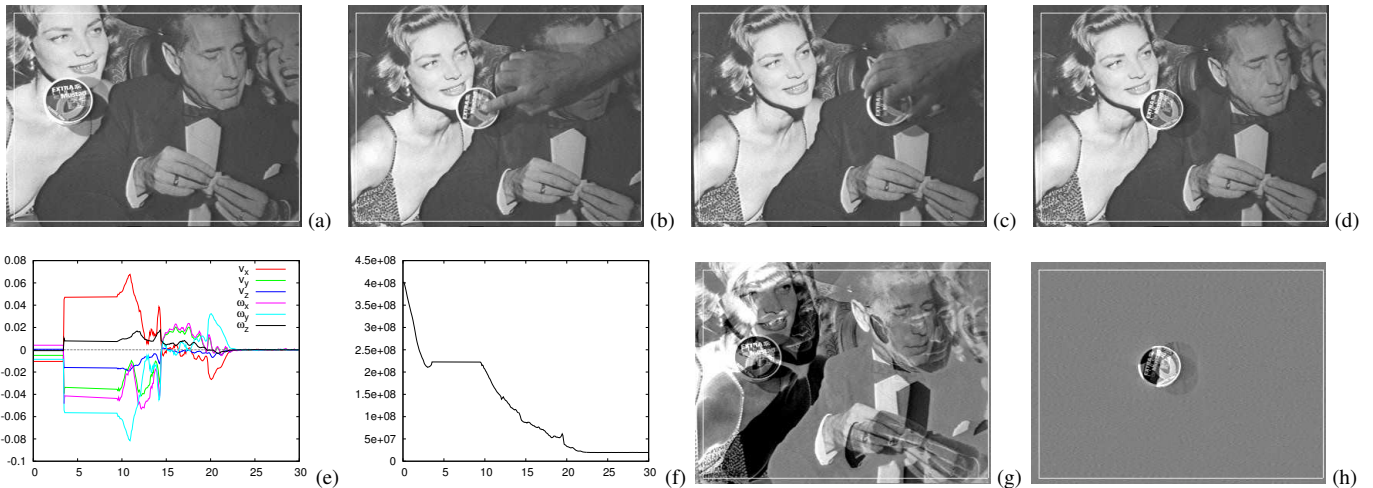


FIG. 7 – Troisième expérimentation. Occultations. Méthode MLM (axes des x en seconde). (a) Image initiale, (b) Image à $t \approx 11$ s, (c) Image à $t \approx 13$ s (d) Image finale, (e) Vitesse de la caméra (m/s ou rad/s), (f) Fonction de coût, (g) $\mathbf{I} - \mathbf{I}^*$ pour la position initiale, (h) $\mathbf{I} - \mathbf{I}^*$ pour la position finale.

les profondeurs. Un objet non plan a donc été utilisé pour cette expérimentation, il est représenté sur la figure 8. De fortes disparités de profondeur sont de ce fait introduites (la hauteur de la tour du château est environ 30 cm). Les poses initiale et finale sont inchangées. La figure 9 décrit cette expérimentation. Ici aussi, la loi de commande converge sans problème et l'erreur de positionnement est toujours très faible, en effet nous avons obtenu $\Delta \mathbf{r} = (0.2 \text{ mm}, -0.2 \text{ mm}, 0.1 \text{ mm}, -0.01^\circ, -0.02^\circ, 0.01^\circ)$.

7 Conclusion et travaux futurs

Nous avons présenté dans cet article l'utilisation de la luminance en tant qu'information visuelle pour l'asservissement visuel. C'est à notre connaissance la première fois qu'une tâche d'asservissement visuel est réalisée sans aucun traitement de l'image (excepté le calcul du gradient spatial de l'image) et sans apprentissage préalable. Pour ce faire, une loi de commande dédiée a été proposée afin de pallier les mauvais comportements des lois de commande utilisées habituellement pour traiter des informations visuelles de nature géométrique. Notre approche a été validée en réalisant diverses tâches de positionnement. L'erreur de positionnement obtenue est toujours très faible. De plus, notre approche s'est révélée être très peu sensible aux occultations partielles de la scène et à de fortes approximations des profondeurs nécessaires pour le calcul de la matrice d'interaction. En outre, même si son calcul a été réalisé à partir d'hypothèses Lambertiennes, les expérimentations ont montré que de très faibles erreurs de positionnement étaient obtenues même dans le cas non Lambertien.

Les travaux futurs concerneront la réalisation de tâche de suivi de cible. Toutefois, puisque la position de l'éclairage ne sera plus constante par rapport à la scène, le calcul de la matrice d'interaction devra être réalisé en prenant en compte de un modèle de réflexion plus réaliste.



FIG. 8 – La scène non plane utilisée dans la dernière expérimentation.

Remerciement

Les auteurs tiennent à remercier Seth Hutchinson pour sa relecture avisée d'une version préliminaire de cet article.

Références

- [1] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. Hager, and A.S. Morse, editors, *The Confluence of Vision and Control*, pages 66–78. LNCIS Series, No 237, Springer-Verlag, 1998.
- [2] 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, Juin 2000.
- [3] K. Deguchi and T. Noguchi. Visual servoing using eigenspace method and dynamic calculation of interaction matrices. In *IEEE Int Conf on Pattern Recognition*, pages 302–306, 1996.
- [4] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on*

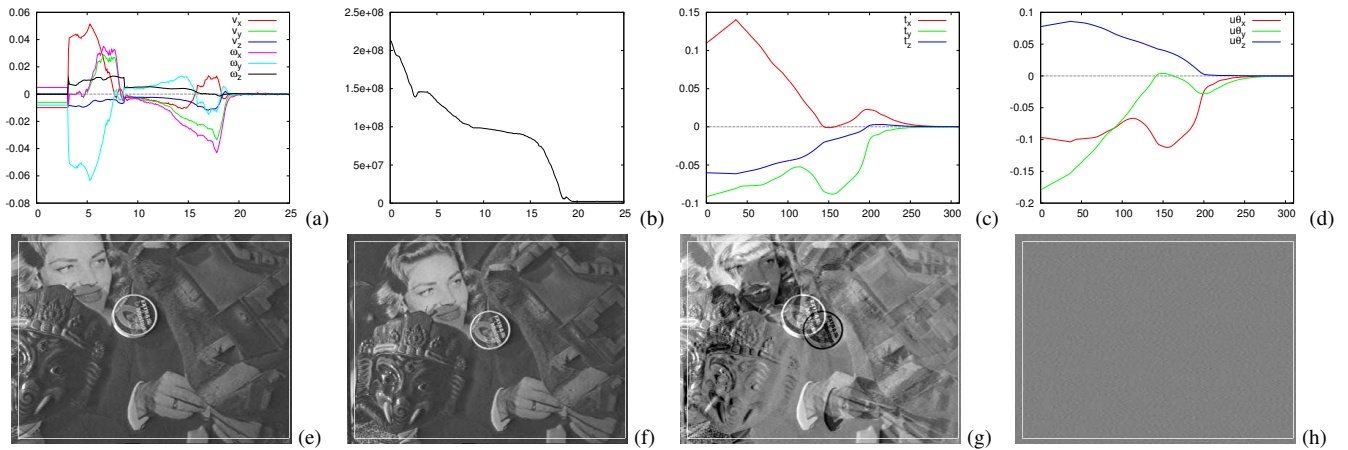


FIG. 9 – Quatrième expérimentation. Robustesse vis-à-vis des profondeurs. Méthode MLM (axes des x en seconde pour (a) et (b), itération pour (c) et (d)). (a) Vitesse de la caméra (m/s ou rad/s), (b) Fonction de coût, (c) Partie translation de $\Delta \mathbf{r}$ (en mètre), (d) Partie rotation de $\Delta \mathbf{r}$ (en radian), (e) Image initiale, (f) Image finale, (g) $I - I^*$ en position initiale, (h) $I - I^*$ en position finale.

Robotics and Automation, 8(3) :313–326, Juin 1992.

- [5] K. Hashimoto and H. Kimura. Lq optimal and non-linear approaches to visual servoing. In K. Hashimoto, editor, *Visual Servoing*, volume 7, pages 165–198. World Scientific Series in Robotics and Automated Systems, Singapour, 1993.
- [6] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3) :185–203, Août 1981.
- [7] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5) :651–670, Octobre 1996.
- [8] V. Kallem, M. Dewan, J.P. Swensen, G.D. Hager, and N.J. Cowan. Kernel-based visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS'07*, San Diego, USA, Octobre 2007.
- [9] J.-T. Lapresté and Y. Mezouar. A Hessian approach to visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS'04*, Sendai, Japon, Octobre 2004.
- [10] E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, volume 2, pages 1843–1848, La Nouvelle-Orléans, Louisiane, Avril 2004.
- [11] E. Marchand. Control camera and light source positions using image gradient information. In *IEEE Int. Conf. on Robotics and Automation, ICRA'07*, Rome, Italie, Avril 2007.
- [12] E. Marchand and F. Chaumette. Feature tracking for visual servoing purposes. *Robotics and Autonomous Systems*, 52(1) :53–70, Juin 2005. special issue on “Advances in Robot Vision”, D. Kragic, H. Christensen (Eds.).
- [13] K. Miura, J. Gangloff, and M. De Mathelin. Robust and uncalibrated visual servoing without Jacobian using a simplex method. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS'02*, Lausanne, Suisse, Octobre 2002.
- [14] S.K. Nayar, S.A. Nene, and H. Murase. Subspace methods for robot vision. *IEEE Trans. on Robotics*, 12(5) :750–758, Octobre 1996.
- [15] G. Silveira and E. Malis. Direct visual servoing with respect to rigid objects. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS'07*, San Diego, USA, Octobre 2007.