

Asservissement visuel direct basé sur des histogrammes d'intensité

Direct visual servoing based on multiple intensity histograms

Quentin Bateux

Eric Marchand

Université de Rennes 1, IRISA, Inria Rennes, France
Quentin.Bateux@irisa.fr, Eric.Marchand@irisa.fr

Résumé

Classiquement, l'asservissement visuel basé image a consisté à contrôler les mouvement d'un ensemble de primitives visuelles (souvent de type géométrique). Récemment, un contrôle de type asservissement visuel photométrique a été proposé, de manière à considérer l'image comme un ensemble et ainsi permettant d'éviter d'extraire et de suivre des primitives géométriques. Des travaux antérieurs ont proposé d'utiliser directement les intensités de l'image pour définir une loi de contrôle. Dans cet article, nous proposons une extension de ces travaux en utilisant des descripteurs globaux, ici des histogrammes d'intensité, pris sur l'intégralité de l'image ou sur plusieurs sous-ensembles de l'image, de manière à arriver à un contrôle des 6 degrés de liberté (ddl.) d'un robot. Les résultats sont ensuite démontrés *via* des validations expérimentales.

Mots Clef

Vision, asservissement visuel direct, histogrammes d'intensités, robotique.

Abstract

Classically visual servoing considered the regulation in the image of a set of visual features (usually geometric features). Recently direct visual servoing scheme, such as photometric visual servoing, have been introduced in order to consider the image as a whole and thus avoid the extraction and the tracking of such geometric features. Previous works proposed methods to use directly the image intensities in the definition of the control law. In this paper, we propose a method to extend these works by using a global descriptor, namely intensity histograms, on the whole or multiple sub-sets of the images in order to achieve control of a 6 degrees of freedom (DoF) robot. The results are then demonstrated through experimental validations.

Keywords

Computer vision, direct visual servoing, intensity histograms, robotics.

1 Introduction

Le but de l'asservissement visuel est de contrôler la dynamique d'un système par l'utilisation de l'information visuelle fournie par une ou plusieurs caméras [2]. Le but est de réguler à zéro une erreur définie dans le plan image. Habituellement, comme présenté dans [11][2][3] cette erreur à minimiser est basée sur des caractéristiques visuelles qui correspondent à des caractéristiques géométriques (points, lignes,...) extraites et mises en correspondance dans l'image courante et désirée. Cette approche classique requiert l'extraction d'informations visuelles qui doivent ensuite être mises en correspondance et suivies à partir de ces images. Le suivi spatio-temporel en temps réel de ces caractéristiques visuelles étant une tâche non triviale, il représente un des points critiques au niveau du développement du domaine de l'asservissement visuel.

Récemment, pour éviter ces processus de suivi et de mise en correspondance, des approches directes ont été proposées [4], dans lesquelles l'étape de mise en correspondance des caractéristiques et d'extraction a été éliminée. Dans le cadre de ces méthodes, le contrôle est construit par minimisation de l'erreur entre l'image courante et désirée. La seule information considérée ici étant l'information photométrique pure (i.e., l'intensité de l'image).

Bien que l'asservissement visuel photométrique [4] soit très efficace et très précis, il peut pâtir d'un certain nombre d'inconvénients dans un environnement non contrôlé (parce qu'il se base sur une comparaison pixel à pixel entre les images, il peut alors être sensible aux variations d'illuminations et aux occultations...), ce qui mène à une perte de précision au niveau des résultats de positionnement. Une possibilité pour améliorer cette méthode est l'utilisation d'une approche globale de manière à travailler avec une représentation plus compacte de l'information visuelle. Pour résoudre ces problèmes, [9] propose d'adapter dynamiquement l'image de référence aux conditions d'illumination de l'image courante acquise par la caméra. L'image de référence est alors remplacée par une image prédite, calculée en utilisant un opérateur de prédiction probabiliste (lequel prend en compte les conditions d'illuminations à la fois de l'image courante et de

l'image désirée apprise). Une autre approche directe [8] a utilisé l'information mutuelle comme critère de similarité entre l'image désirée et courante. Cette approche basée sur la théorie de l'information permet d'asservir la caméra de manière à maximiser la quantité d'information partagée entre l'image courante et désirée (l'information mutuelle étant basée sur une mesure de l'entropie de l'image).

Dans cet article, nous proposons de représenter l'image par des histogrammes d'intensité et de construire une loi de contrôle basée sur ces caractéristiques visuelles, ce qui permet d'offrir une plus grande flexibilité au niveau du choix du descripteur (niveaux de gris, couleur, gradients...) et permettant de profiter de la relative simplicité de calcul d'une distance entre histogrammes. Les histogrammes ont été très largement utilisés en vision par ordinateur, que ce soit en suivi, mise en correspondance, détection ou indexation d'images.

Au niveau du suivi, l'algorithme de suivi *Mean Shift* [6] et ses développements ont prouvé la faisabilité d'un schéma de commande basé histogramme. Un histogramme de couleur est utilisé pour décrire la région à suivre. En se basant sur le coefficient de Bhattacharyya, le suivi en ensuite accompli itérativement pour trouver le minimum local d'une fonction de mesure de distance en utilisant l'algorithme du mean shift. Une version étendue du mean shift, utilisant des histogrammes et basé sur de l'optimisation non-linéaire, a aussi été proposée dans [10]. Dans le domaine de la reconnaissance de forme ou de l'indexation d'image, l'utilisation de méthodes utilisant des descripteurs basés histogrammes a bénéficié grandement à certaines applications. Les histogrammes de gradients orientés ont été largement utilisés dans la détection de points d'intérêts et dans les méthodes de mise en correspondance telles que Sift [12], utilisé par exemple dans la détection de piétons comme montré dans [7]. Des algorithmes d'indexation ou de recherche d'image, tels que *Gist* [14], utilisent eux aussi les histogrammes pour décrire l'image globalement.

Dans cet article, nous étendons l'asservissement visuel direct en utilisant les histogrammes d'intensités de niveaux de gris. L'idée est de calculer l'histogramme des images acquises pour la pose désirée et courante, et de commander la caméra de manière à minimiser une distance appropriée entre ces deux histogrammes. Cette approche nous permet de bénéficier des propriétés des histogrammes, telles qu'une meilleure robustesse au bruit, ainsi que de la flexibilité en permettant de choisir le nombre de classes considérées dans le calcul de l'histogramme.

Cet article est structuré comme suit. La section suivante décrit les concepts élémentaires de l'asservissement visuel. La section 3 développe la nouvelle fonction de coût à minimiser et la loi de commande sous-jacente. Finalement, la section 4 propose le développements expérimentaux qui valident la méthode.

2 Asservissement visuel direct

2.1 Tâche de positionnement

L'objectif d'une tâche de positionnement est d'atteindre une pose désirée de la caméra \mathbf{r}^* , en partant d'une pose initiale arbitraire. Pour atteindre cet objectif, il faut définir une fonction de coût ρ qui reflète, dans l'espace image, cette erreur. En considérant la pose courante de la caméra \mathbf{r} , le problème peut alors être écrit sous la forme d'un processus d'optimisation :

$$\hat{\mathbf{r}} = \arg \min_{\mathbf{r}} \rho(\mathbf{r}, \mathbf{r}^*) \quad (1)$$

où $\hat{\mathbf{r}}$ est la pose atteinte après le processus d'asservissement, et doit être la plus proche possible de \mathbf{r}^* (à l'optimum $\hat{\mathbf{r}} = \mathbf{r}^*$). Par exemple, si l'on considère un ensemble de caractéristiques géométriques \mathbf{s} , de manière classique, la tâche minimisera la différence entre \mathbf{s} et la configuration désirée \mathbf{s}^* , ce qui mène à :

$$\hat{\mathbf{r}} = \arg \min_{\mathbf{r}} (\mathbf{s}(\mathbf{r}) - \mathbf{s}^*). \quad (2)$$

Cette tâche d'asservissement visuel peut être résolue en appliquant itérativement une vitesse à la caméra. Cela nécessite une connaissance de la matrice d'interaction \mathbf{L}_s de $\mathbf{s}(\mathbf{r})$ qui relie la variation de $\dot{\mathbf{s}}$ à la vitesse de la caméra et est définie telle que :

$$\dot{\mathbf{s}}(\mathbf{r}) = \mathbf{L}_s \mathbf{v} \quad (3)$$

avec $\mathbf{v} = (v, \omega)$, où v et ω sont respectivement la vitesse linéaire et angulaire de la caméra.

Cette équation mène à l'expression de la vitesse qu'il faut appliquer au robot. Une loi de commande cinématique est traditionnellement définie par :

$$\mathbf{v} = -\lambda \mathbf{L}_s^+ (\mathbf{s}(\mathbf{r}) - \mathbf{s}^*) \quad (4)$$

où λ est un scalaire positif.

Comme indiqué dans [5], une telle loi de commande est très similaire à un processus de minimisation de type Gauss-Newton, méthode très largement utilisée pour résoudre des problèmes d'optimisation non-linéaires. Dans cette approche, la détermination de la direction de descente va affecter très largement le rayon de convergence de cette méthode. Une méthode alternative, de type Levenberg-Marquardt, peut aussi être utilisée afin d'augmenter le cône de convergence et peut s'exprimer de la manière suivante :

$$\mathbf{v} = -(\mathbf{H} + \mu \text{diag}(\mathbf{H}))^{-1} \mathbf{L}_s^\top (\mathbf{s}(\mathbf{r}) - \mathbf{s}^*) \quad (5)$$

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

3 Asservissement visuel basé histogramme

Les histogrammes sont un moyen très puissant d'obtenir une représentation globale et compacte d'un ensemble

de valeurs, spécialement pour des applications de suivi ou d'indexation d'images, où ils sont utilisés pour effectuer des comparaisons entre deux images (ou des parties d'images) de manière globale. Puisque la comparaison pixel à pixel (SSD) peut ne pas être assez robuste en cas de variation d'illumination ou de bruit, une telle approche globale nous apparaît être un choix intéressant.

3.1 Histogramme : définition et présentation

L'idée principale derrière le calcul d'histogramme est de classifier chaque pixel de l'image dans une classe correspondant (dans le cas considéré dans cet article) à une valeur d'intensité de ce pixel. Chaque classe consiste en une valeur scalaire qui sera incrémentée chaque fois qu'un élément correspondant à cette classe sera trouvé. Les valeurs des classes seront ensuite normalisées en fonction du nombre de pixels dans l'image. Cela conduit à la définition classique d'un histogramme d'intensité :

$$p_{\mathbf{I}}(i) = \frac{1}{N_{\mathbf{x}}} \sum_{\mathbf{x}} \delta(\mathbf{I}(\mathbf{x}) - i) \quad (6)$$

où $i \in [0, 255]$ si l'on utilise des images à 256 niveaux de gris, $N_{\mathbf{x}}$ est le nombre de pixels dans l'image $\mathbf{I}(\mathbf{x})$, et $\delta(\mathbf{I}(\mathbf{x}) - i)$ étant la fonction de Kronecker définie par :

$$\delta(x) = \begin{cases} 1 & \text{si } x = 0 \\ 0 & \text{sinon} \end{cases} \quad (7)$$

$p_{\mathbf{I}}(i)$ est donc la probabilité qu'un pixel de l'image \mathbf{I} possède l'intensité i .

3.2 Distance entre histogrammes

Comme annoncé, notre but est de trouver un moyen d'utiliser un tel histogramme en tant que caractéristique visuelle au sein d'une loi de commande d'asservissement visuel. Il convient donc de pouvoir comparer deux à deux ces histogrammes. Comme donné dans l'équation (1), une fonction de corrélation $\rho(\cdot)$ doit donc être définie. La méthode traditionnelle pour le faire est de comparer les histogrammes classe à classe, en utilisant par exemple la distance de Matusita, qui peut être exprimée par [1] :

$$\rho(\mathbf{I}, \mathbf{I}^*) = \sum_i^{N_c} \left(\sqrt{p_{\mathbf{I}}(i)} - \sqrt{p_{\mathbf{I}^*}(i)} \right)^2 \quad (8)$$

où N_c est le nombre de classes utilisées dans les histogrammes. Il est à noter que le choix d'un nombre de classes élevé permettra d'obtenir plus de précision au niveau de la loi de commande mais au détriment du rayon de la zone de convergence et peut aussi conduire à l'apparition de minima locaux dans la fonction de coût ce qui peut nuire à la convergence de la loi de commande.

3.3 Calcul de la matrice d'interaction

Le problème principal qui survient lors de l'utilisation de cette formulation de l'histogramme est que la fonction

de Kronecker ($\delta(\cdot)$ dans l'équation (6)) est a priori non-dérivable, ce qui est problématique puisque nous devons calculer une matrice d'interaction [2] (dite aussi Jacobien image [11]) correspondant à cette nouvelle fonction d'erreur (à savoir la distance de Matusita).

Pour résoudre ce problème, en suivant la méthodologie proposée dans [8], nous choisissons d'utiliser des B-Splines d'ordre au moins deux (dérivable au moins une fois) pour approcher et adoucir nos classes (comme illustré par la Fig.1). Cette technique conduit à la formulation suivante d'un histogramme :

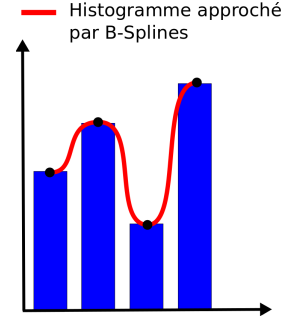


FIGURE 1 – B-Spline appliquée à un histogramme

$$p_{\mathbf{I}}(i) = \frac{1}{N_{\mathbf{x}}} \sum_{\mathbf{x}} \phi(\bar{\mathbf{I}}(\mathbf{x}) - i) \quad (9)$$

où $N_{\mathbf{x}}$ est le nombre de pixels, $\mathbf{x} = \{x, y\}$ est un unique pixel de l'image, $\phi(\cdot)$ est une B-Spline dérivable au moins une fois, $\bar{\mathbf{I}}$ est l'image réduite à N_c valeurs d'intensité (chaque histogramme contiendra alors N_c classes). Pour des images 8 bits, nous avons :

$$\bar{\mathbf{I}}(\mathbf{x}) = \frac{N_c}{256} \mathbf{I}(\mathbf{x}).$$

Une description détaillée des fonctions B-Splines est donnée par Unser *et al.* dans [15], mais les propriétés qui nous intéressent ici sont : l'intégrale de la fonction est 1 (ainsi le résultat n'a pas à être re-normalisé) et le calcul des dérivées est obtenu aisément. Dans les expériences suivantes, une B-Spline d'ordre 2 a été choisie :

$$\phi(t) = \begin{cases} t + 1 & \text{si } t \in [-1, 0] \\ -t + 1 & \text{si } t \in [0, 1] \\ 0 & \text{sinon} \end{cases} \quad (10)$$

Comme montré dans la partie précédentes, la matrice d'interaction peut être vue comme le Jacobien du vecteur de caractéristiques par rapport à chacun des paramètres de pose $\mathbf{r} = \{t_x, t_y, t_z, r_x, r_y, r_z\}$. Cette définition mène à l'expression suivante : $\mathbf{L}_{\rho} = \frac{\partial \rho(\mathbf{I}, \mathbf{I}^*)}{\partial \mathbf{r}}$ où \mathbf{I}^* et \mathbf{I} sont les images à la pose désirée et à la pose courante, et $\rho(\mathbf{I}, \mathbf{I}^*)$ est la distance de Matusita entre les deux histogrammes.

En utilisant la règle de composition des dérivés de fonctions, on peut obtenir les expressions suivantes :

$$\begin{aligned} \mathbf{L}_\rho &= \frac{\partial}{\partial \mathbf{r}} \left[\sum_i^{N_c} \left(\sqrt{\mathbf{P}_{\bar{\mathbf{I}}}(i)} - \sqrt{\mathbf{P}_{\bar{\mathbf{I}}^*}(i)} \right)^2 \right] \\ &= 2 \sum_i^{N_c} \left(\frac{\partial}{\partial \mathbf{r}} \sqrt{\mathbf{P}_{\bar{\mathbf{I}}}(i)} \left(\sqrt{\mathbf{P}_{\bar{\mathbf{I}}}(i)} - \sqrt{\mathbf{P}_{\bar{\mathbf{I}}^*}(i)} \right) \right) \end{aligned} \quad (11)$$

avec

$$\frac{\partial}{\partial \mathbf{r}} \sqrt{\mathbf{P}_{\bar{\mathbf{I}}}(i)} = \frac{1}{2\sqrt{\mathbf{P}_{\bar{\mathbf{I}}}(i)}} \frac{\partial \mathbf{P}_{\bar{\mathbf{I}}}(i)}{\partial \mathbf{r}} \quad (12)$$

où

$$\begin{aligned} \frac{\partial \mathbf{P}_{\bar{\mathbf{I}}}(i)}{\partial \mathbf{r}} &= \frac{\partial}{\partial \mathbf{r}} \left(\sum_{\mathbf{x}}^{N_x} (\phi(\bar{\mathbf{I}}(\mathbf{r}, \mathbf{x}) - i)) \right) \\ &= \sum_{\mathbf{x}}^{N_x} \left(\frac{\partial}{\partial \mathbf{r}} (\phi(\bar{\mathbf{I}}(\mathbf{r}, \mathbf{x}) - i)) \right) \end{aligned} \quad (13)$$

et d'après [8],

$$\begin{aligned} \frac{\partial}{\partial \mathbf{r}} (\phi(\bar{\mathbf{I}}(\mathbf{r}, \mathbf{x}) - i)) &= \frac{\partial}{\partial i} (\phi(\bar{\mathbf{I}}(\mathbf{r}, \mathbf{x}) - i)) \frac{\partial \bar{\mathbf{I}}(\mathbf{r}, \mathbf{x})}{\partial \mathbf{r}} \\ &= \frac{\partial}{\partial i} (\phi(\bar{\mathbf{I}}(\mathbf{r}, \mathbf{x}) - i)) \mathbf{L}_{\bar{\mathbf{I}}} \end{aligned} \quad (14)$$

avec, d'après [5], la matrice d'interaction qui relie la variation des intensités de l'image à la vitesse de la caméra, donnée par :

$$\mathbf{L}_{\bar{\mathbf{I}}} = -(\nabla_{\mathbf{x}} \bar{\mathbf{I}} \mathbf{L}_{\mathbf{x}} + \nabla_{\mathbf{y}} \bar{\mathbf{I}} \mathbf{L}_{\mathbf{y}}) \quad (15)$$

où $\mathbf{L}_{\mathbf{x}}$ et $\mathbf{L}_{\mathbf{y}}$ sont les lignes correspondant aux coordonnées x et y de $\mathbf{L}_{\mathbf{x}}$ qui est la matrice d'interaction correspondant à un unique point 2D $\mathbf{x} = (x, y)$ de l'image, et définie dans [2] par :

$$\mathbf{L}_{\mathbf{x}} = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix}.$$

Finalement, après simplification, nous obtenons la matrice d'interaction suivante \mathbf{L}_ρ pour notre loi de commande :

$$\mathbf{L}_\rho = \sum_i^{N_c} \left(\frac{1}{N_x} \left(1 - \frac{\sqrt{\mathbf{P}_{\bar{\mathbf{I}}^*}(i)}}{\sqrt{\mathbf{P}_{\bar{\mathbf{I}}}(i)}} \right) \sum_{\mathbf{x}}^{N_x} \left(\frac{\partial}{\partial i} (\phi(\bar{\mathbf{I}}(\mathbf{r}, \mathbf{x}) - i)) \mathbf{L}_{\bar{\mathbf{I}}} \right) \right).$$

3.4 Utilisation de noyaux dans l'histogramme

Dans des travaux tels que les problèmes de suivi par mean shift [6, 10], lesquels sont liés à notre problématique, une fonction noyau de pondération a été introduite avec succès dans le calcul de l'histogramme, de manière à donner plus de poids aux pixels au centre de l'image et moins à ceux en bordure, ces derniers étant plus susceptibles de contenir de l'information utile. L'utilisation des noyaux peut être justifiée puisque certains pixels entrent et sortent du champ

de capture de la caméra. Le noyau permet alors de donner plus d'importance à ceux, plus stables, du centre de l'image. L'expression de calcul de l'histogramme devient alors :

$$\mathbf{P}_{\bar{\mathbf{I}}}(i) = C \sum_{\mathbf{x}}^{N_x} K\left(\frac{\mathbf{x} - \mathbf{c}}{h}\right) \delta(i - \bar{\mathbf{I}}(\mathbf{x})) \quad (16)$$

avec C une constante de normalisation telle que $C = \frac{1}{N_x \sum_{\mathbf{x}}^{N_x} K\left(\frac{\mathbf{x} - \mathbf{c}}{h}\right)}$, et K un noyau isotrope de bande passante h , centré sur \mathbf{c} , le centre de l'image. Dans notre cas, nous prenons K comme étant un noyau gaussien, exprimé tel que :

$$K\left(\frac{\mathbf{x} - \mathbf{c}}{h}\right) = e^{-\frac{\|\mathbf{x} - \mathbf{c}\|_2^2}{h}}. \quad (17)$$

3.5 Extension à des noyaux multiples

Comme attendu d'après des travaux précédents utilisant des distances d'histogrammes d'intensité [6], le calcul d'un seul histogramme sur l'image entière ne permet pas de contrôler plus que les seules translations sur les axes x/y . Une solution qui a été proposée dans le domaine du suivi est d'étendre la sensibilité de la méthode basée histogramme à plus de degrés de libertés en utilisant plusieurs noyaux dans la même image [10]. Ici nous adaptions cette idée en divisant l'image en plusieurs zones et en associant un histogramme à chacune de ces zones, les vecteurs d'erreurs ainsi que les matrices d'interaction résultantes sont ensuite empilés. La Fig. 2 montre un exemple de l'utilisation de cette méthode en utilisant 9 noyaux dans l'image : chaque sous-partie est utilisée pour calculer son propre histogramme, donnant ainsi une nouvelle distance qui pourra être utilisée comme une caractéristique à part entière lorsque l'on comparera ces 9 histogrammes aux 9 histogrammes qui seront calculés dans l'image à la pose désirée.

La matrice d'interaction globale devient alors :

$$\mathbf{L}_\rho = [\mathbf{L}_{\rho_1} \quad \mathbf{L}_{\rho_2} \quad \dots \quad \mathbf{L}_{\rho_n}]^T \quad (18)$$

où \mathbf{L}_{ρ_i} est la matrice d'interaction associée à la distance entre les K -ièmes histogrammes, en utilisant la loi de commande définie dans la partie précédente. En utilisant assez de noyaux dans l'image, il devient alors possible de contrôler les six degrés de liberté du robot (voir la partie suivante).

Après plusieurs expériences, la séparation de l'image en cinq horizontalement et verticalement (25 sous-parties) s'est avérée être un bon choix. Prendre moins de zones peut se révéler problématique à par rapport à un bon découplage des degrés de liberté, tandis qu'en prendre plus peut entraîner une trop grande sensibilité à de petites variations de la scène et conduire à une plus grande difficulté concernant le réglage des paramètres (notamment concernant la détermination du gain associé à la vitesse de descente et du nombre de classes à prendre dans l'histogramme).



FIGURE 2 – Illustration de l’approche multi-noyau avec 2 séparations horizontales et verticales (9 zones)

Concernant l’effet de l’approche multi-noyaux au niveau de la fonction de coût, la Fig. 3 montre la forme de la fonction de coût dans le cas où l’on utilise qu’un seul noyau sur toute l’image pour calculer notre histogramme. On voit que même sur ces seuls degrés de libertés (t_x et t_y), il n’y a pas de minimum global clairement défini, ce qui pourra empêcher notre système d’atteindre la pose désirée. D’un autre côté, si l’on sépare l’image en 5 sur les axes x et y, et si l’on crée une caractéristique pour chacune des sous-parties, la fonction de coût montre un minimum bien défini, ainsi qu’une vallée de convergence plutôt large, comme le montre la Fig. 4.

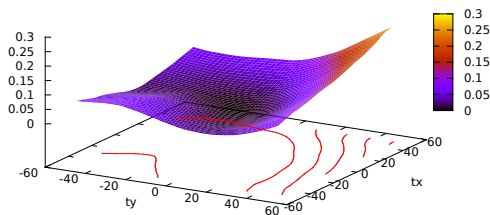


FIGURE 3 – Fonction de coût pour une approche à noyau simple

4 Résultats expérimentaux

Pour valider l’approche proposée, plusieurs expériences ont été effectuées en utilisant une caméra montée sur un bras robotisé à 6 degrés de liberté. Indépendamment de l’expérience, le temps de calcul reste faible. La loi de commande est calculée à la cadence vidéo. Une vitesse est calculée et envoyée au robot toutes les 40ms pour une image d’entrée de taille 320×240 sur un ordinateur de cadence 2.4GHz. Toutes les implémentations ont été effectuées en

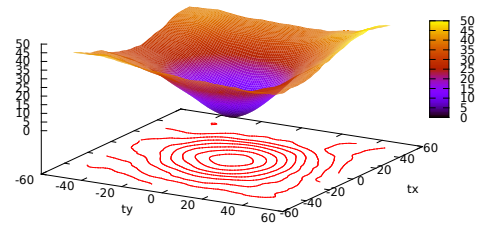


FIGURE 4 – Fonction de coût pour une approche à 25 noyaux

utilisant la librairie ViSP [13].

Pour la validation de notre approche, plusieurs expériences ont été mises en place pour tester l’efficacité globale de cette méthode. Le but de ces expériences est toujours le même, nous indiquons une pose désirée au robot en captant l’image associée, puis nous déplaçons le robot à une autre pose, ensuite, les vitesses à appliquer à la caméra sont calculées à la cadence vidéo selon la loi de commande proposée de manière à revenir à la pose désirée. Quatre expériences ont été réalisées, testant la méthode en conditions nominales (avec un robot à 4 ddl. lors d’une simulation préliminaire et avec 6 ddl. sur le robot physique), dans le cas d’une scène non plane et enfin en présence d’occlusions qui sont ajoutées durant l’expérience.

Simulation d’une tâche de positionnement à 4 ddl.

Pour la première expérience, pour valider la loi de commande proposée, nous allons considérer une tâche de positionnement à 4 ddl. dans un environnement simulé. L’image simulée est un environnement plan, fourni par le moteur de rendu de ViSP [13]. Dans ce cas de test, la caméra est positionnée à une distance de 0.2m du plan objet, avec un décalage initial au niveau de la position caméra de $r_{init} = (1cm, -1cm, -2.5cm, 0^\circ, 0^\circ, 10^\circ)$. Comme montré sur la Fig. 5, la loi de commande réussit à converger, ouvrant la voie à des tests plus complexes d’asservissement visuel sur robot physique à 6 ddl.

Tâche de positionnement à 6 ddl. en conditions nominales.

Pour la deuxième expérience, nous avons travaillé en conditions nominales, c’est à dire avec une scène plane, sans modification de la luminosité et sans apparition d’éléments extérieurs à la scène initiale durant l’expérience. La profondeur a été considérée comme étant constante durant toutes les expériences, avec $Z = 0.80m$ dans l’équation (16). Le déplacement initial par rapport à la position désirée (qui rend le plan caméra parallèle au plan de la scène) choisi pour cette expérience était de $r_{init} = (-2cm, 16cm, 6cm, 16^\circ, 1^\circ, 0^\circ)$. La Fig. 6 montre les résultats de cette expérience. Il est intéressant de noter qu’avec cette pose initiale, le plan de la scène et celui de la caméra ne sont plus parallèles, ce qui entraîne une profondeur relative non-uniforme des pixels de l’image ac-

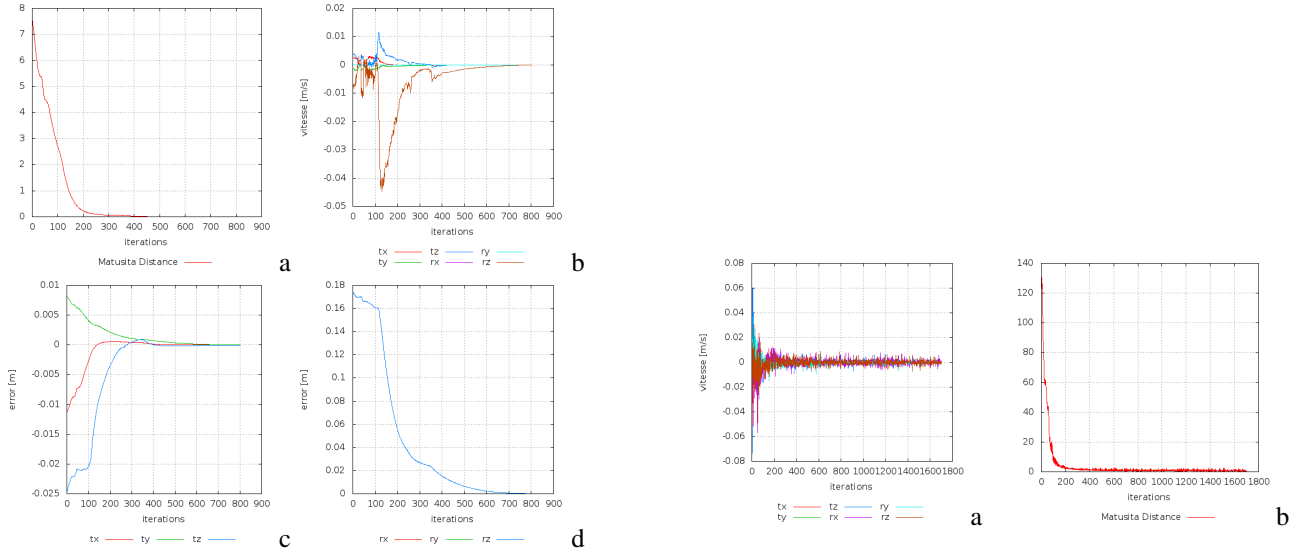


FIGURE 5 – Simulation à 4 ddl. Vitesses caméra en m/s et rad/s dans (a). (b) Distance de Matusita. (c) Erreur de positionnement : composantes de translations de Δr , (d) composantes de rotations de Δr .

quise. Malgré l'hypothèse forte de profondeur constante utilisée dans le calcul de la matrice d'interaction, le comportement est convergent. La Fig. 6(a) décrit l'évolution des vitesses de la caméra ; la Fig. 6(b) l'évolution de la distance de Matusita ; les Fig. 6(c) et Fig. 6(d) l'erreur de pose (resp. les composantes translationnelles et rotationnelles) entre r et r^* ; les Fig. 6(e), Fig. 6(f) montrent les images initiales et finales acquises par la caméra ; les Fig. 6(g), Fig. 6(h) décrivent les valeurs initiales de $I - I^*$. La convergence possède un certain nombre de bonne propriétés au niveau de l'évolution de la distance de Matusita, montrant une décroissance exponentielle satisfaisante, laquelle peut aussi être retrouvée dans l'évolution des composantes d'erreurs de pose.

Scène non plane. Dans cette expérience, nous testons la sensibilité qui peut être induite par notre hypothèse de planarité de la scène en introduisant un objet 3D dans notre scène. L'objet utilisé peut être vu sur les Fig 7(a) et 7(b). La pose initiale choisie pour cette expérience était $r_{init} = (-1cm, 0cm, -10cm, 0^\circ, 0^\circ, -19^\circ)$, ce qui permet aussi de tester la sensibilité introduite par l'hypothèse de profondeur constante dans le calcul de la matrice d'interaction. La Fig. 7 montre le résultat de l'expérience, et nous pouvons voir que toutes les bonnes propriétés de l'expérience précédente sont conservées, démontrant ainsi la robustesse de notre méthode concernant les approximations effectuées sur la profondeur. Pour cette expérience, nous avons utilisé uniquement 16 classes dans notre histogramme, ce qui a pour conséquence d'élargir la zone de convergence.

Ajout d'occlusions durant le positionnement. Dans cette dernière expérience, nous introduisons un objet externe à la scène durant le processus de positionnement.

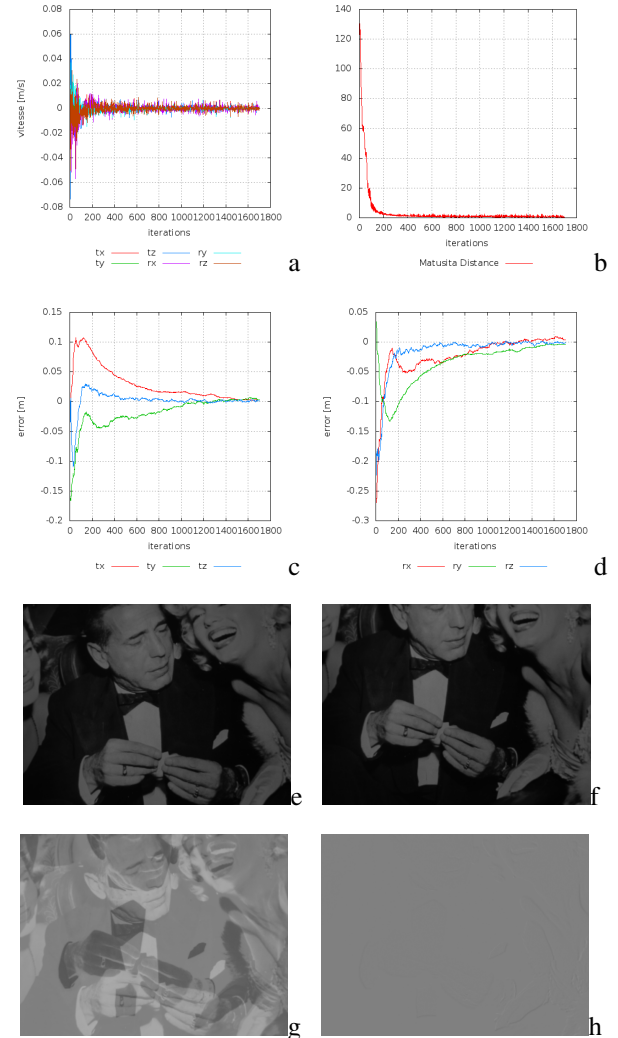


FIGURE 6 – Conditions nominales. Vitesses caméra en m/s et rad/s dans (a). (b) Distance de Matusita. (c) Erreur de positionnement : composantes de translations de Δr , (d) composantes de rotations de Δr . (e) Image initiale (128 niveaux de gris utilisés). (f) Image désirée (128 niveaux de gris utilisés). (g) $I - I^*$ à la position initiale. (h) $I - I^*$ à la fin de l'évolution

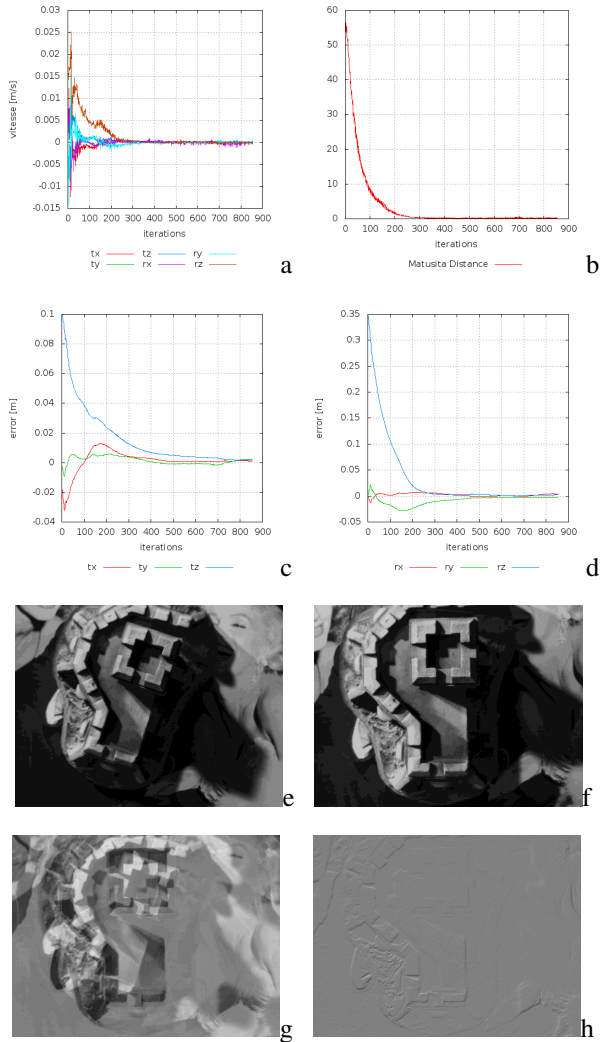


FIGURE 7 – Positionnement par rapport à une scène non plane. Vitesses caméra en m/s et rad/s dans (a). (b) Distance de Matusita. (c) Erreur de positionnement : composantes de translations de Δr , (d) composantes de rotations de Δr . (e) Image initiale (16 niveaux de gris utilisés). (f) Image désirée (16 niveaux de gris utilisés). (g) $I - I^*$ à la position initiale. (h) $I - I^*$ à la fin de l'évolution

Cela introduit une dissimilarité significative entre l'image désirée, et celle qui peut effectivement être atteinte. Puisque que cela introduit un changement important dans les intensités de nombreux pixels, cela montre aussi que la méthode proposée est robuste aux changements locaux. La pose initiale choisie dans cette expérience était $r_{init} = (-2cm, 16cm, 6cm, 16^\circ, 1^\circ, 0^\circ)$, la même que celle utilisée dans la deuxième expérience, de manière à pouvoir comparer les comportements. La Fig. 8(b) met en valeur le moment où nous ajoutons un objet à la scène. Cet objet peut être clairement vu dans la Fig. 8(f) et Fig. 8(h). Nous pouvons voir que le système converge toujours, malgré des vitesses et des trajectoires plus bruitées.

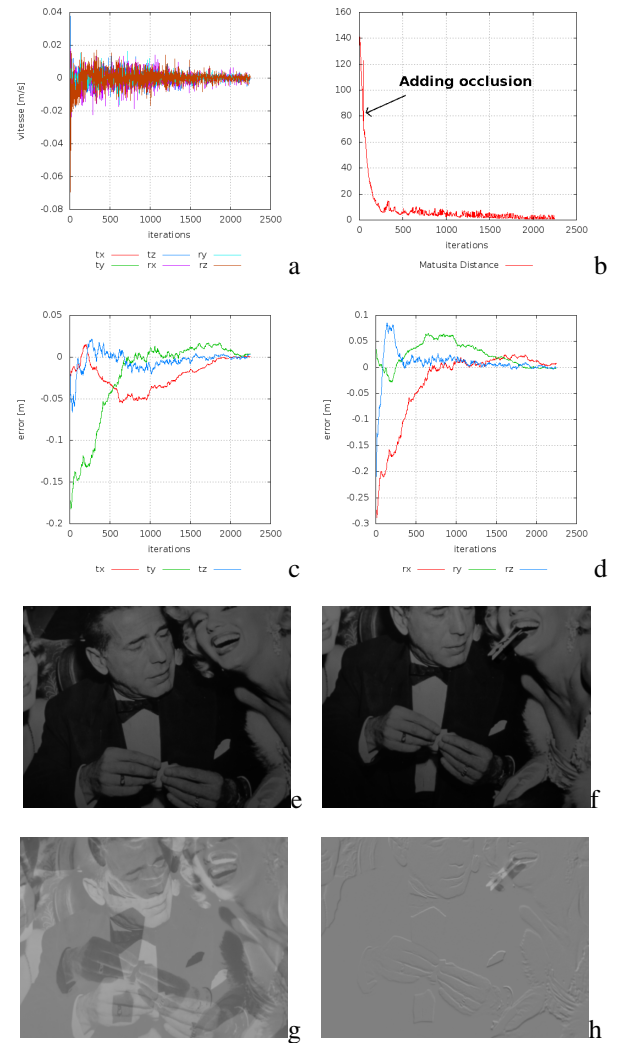


FIGURE 8 – Oclusions pendant le déplacement caméra. Vitesses caméra en m/s et rad/s dans (a). (b) Distance de Matusita. (c) Erreur de positionnement : composantes de translations de Δr , (d) composantes de rotations de Δr . (e) Image initiale (128 niveaux de gris utilisés). (f) Image désirée (128 niveaux de gris utilisés). (g) $I - I^*$ à la position initiale. (h) $I - I^*$ à la fin de l'évolution

Augmenter la zone de convergence. De manière à augmenter la zone de convergence en cas de bruits plus importants tels que des changements d'illuminations ou des occlusions de plus grande taille, les expériences ont montré qu'utiliser un filtre Gaussien au cours d'une phase de prétraitement des images permet une stabilisation significative de la méthode dans la plupart des cas, au dépend toutefois de la précision finale. Toutefois, cela pourrait se révéler intéressant si cette solution était couplée à notre technique actuelle, basée sur des images brutes, au sein d'une approche comportant deux étapes, où une première étape qui amènerait le système dans la zone de convergence grâce à un tel filtre, pour ensuite revenir aux images brutes de manière à obtenir une bonne précision finale.

5 Conclusion et perspectives

Dans cet article, nous avons montré qu'il était possible d'étendre la méthode d'asservissement visuel direct photométrique en utilisant des descripteurs globaux : les histogrammes d'intensité. Nous avons proposé une nouvelle loi de commande minimisant la distance de Matusita entre histogrammes. La matrice d'interaction associée à cette nouvelle fonction d'erreur a été décrite. Enfin nous avons validé cette approche par des tâches de positionnement sur 6 ddl., effectuées sur des scènes variées, à la fois planaires, et non planaires. Les travaux futurs concerneront l'élaboration de lois de contrôle basées sur des descripteurs globaux plus complexes et plus robustes tels que des histogrammes de couleurs, ou des histogrammes de gradients orientés [7].

Références

- [1] S.-H. Cha and S. Srihari. On measuring the distance between histograms. *Pattern Recognition*, 35(6) :1355–1370, 2002.
- [2] F. Chaumette and S. Hutchinson. Visual servo control, Part I : Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4) :82–90, December 2006.
- [3] F. Chaumette and S. Hutchinson. Visual servo control, Part II : Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1) :109–118, March 2007.
- [4] C. Collewet and E. Marchand. Photometric visual servoing. *IEEE Trans. on Robotics*, 27(4) :828–834, August 2011.
- [5] C. Collewet, E. Marchand, and F. Chaumette. Asservissement visuel base sur des informations photométriques. In *16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle, RFIA'08*, Amiens, France, January 2008.
- [6] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on PAMI*, 25(5) :564–577, May 2003.
- [7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'05*, pages 886–893, 2005.
- [8] A. Dame and E. Marchand. Mutual information-based visual servoing. *IEEE Trans. on Robotics*, 27(5) :958–969, October 2011.
- [9] B. Delabarre and E. Marchand. Visual servoing using the sum of conditional variance. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'12*, pages 1689–1694, Vilamoura, Portugal, October 2012.
- [10] G. Hager, M. Dewan, and C. Stewart. Multiple kernel tracking with ssd. In *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'04*, pages 790–797, June 2004.
- [11] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5) :651–670, October 1996.
- [12] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2) :91–110, 2004.
- [13] E. Marchand and F. Chaumette. Feature tracking for visual servoing purposes. *Robotics and Autonomous Systems*, 52(1) :53–70, June 2005. special issue on “Advances in Robot Vision”, D. Kragic, H. Christensen (Eds.).
- [14] A. Oliva and A. Torralba. Building the gist of a scene : The role of global image features in recognition. *Progress in brain research*, 155 :23–36, 2006.
- [15] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing : Part i-theory. *IEEE Trans. Signal Processing*, 41(2) :821–833, February 1993.