Congrès francophone de Vision par Ordinateur, ORASIS'03
Gerardmer, France, Mai 2003

Suivi robuste d'objets complexes en temps réel par asservissement visuel virtuel

Real-time robust and complex object tracking by virtual visual servoing

Andrew I. Comport Éric Marchand François Chaumette

IRISA - INRIA Rennes
Campus de Beaulieu, 35042 Rennes, France

Prénom.Nom@irisa.fr

Résumé

La suivi d'objet complexe en temps réel pour un système de vision monoculaire s'effectue ici en se basant sur un modèle 3D de l'objet. Ces travaux utilisent les principes de l'asservissement visuel virtuel. Dans ce contexte, des matrices d'interaction point-à-courbes sont déterminées afin de minimiser la distance entre des points locaux et les contours auxquels ils appartiennent. Un suivi local en temps réel est obtenu par l'utilisation de la méthode des Éléments de Contour en Mouvement (ECM), laquelle effectue un suivi sur la normale du contour. La robustesse est assurée en intégrant une M-estimation dans la loi de commande, par une méthode des moindres carrés pondérés itérés. L'approche a été validée sur des applications d'asservissement visuel et de réalité augmentée. Plusieurs séquences d'images complexes sont considérées, y compris pour des environnements extérieurs. Les résultats obtenus montrent la robustesse de la méthode aux occultations, avec des performances de traitement satisfaisantes.

Mots Clés

Suivi 3D, Temps réel, CAO, Réalité augmentée

Abstract

This paper proposes a real-time, robust and efficient 3D model-based tracking algorithm for a monocular vision system. Non-linear pose computation is formulated by means of a virtual visual servoing approach. In this context, the derivation of point-to-curves interaction matrices are given for different features. A local moving edges tracker is used in order to provide real-time tracking of points normal to the object contours. Robustness is obtained by integrating a M-estimator into the visual control law via an iteratively re-weighted least squares implementation. The method presented in this paper has been validated on several complex image sequences including outdoor environments. Furthermore, it is verified with visual servoing as well as augmented reality applications. Results show the method to be robust to occlusion, changes in illumination and miss-tracking.

Keywords

3D Tracking, Real-time, CAD, Augmented Reality

1 Introduction

Cet article traite du problème du suivi 3D d'objets en temps réel dans des séquences d'image monoculaires. Ce problème, fondamental en vision par ordinateur, a des applications dans des domaines tels que la réalité augmentée, l'asservissement visuel et même cértaines applications médicales. Le suivi 3D peut être considéré comme un asservissement visuel 2D pour une caméra virtuelle dont la position initiale est obtenue à partir de l'image précédente ou d'une position initiale proche [21, 19]. Dans cet article, nous présentons un nouvel algorithme d'asservissement visuel virtuel qui aborde les aspects d'efficacité, stabilité, robustesse et de précision. En particulier, ce principe fournit une méthode générale pour la dérivation des matrices d'interaction pour des primitives complexes incluant des ellipses, cylindres, points, distances ou toute autre combinaison.

L'avantage principal des méthodes basées sur un modèle 3D de l'objet est que la connaissance a priori sur la scène (l'information 3D implicite) permet l'amélioration de la robustesse et de la performance tout en étant capable de fournir l'information supplémentaire nécessaire pour réduire les effets de données aberrantes présentes éventuellement dans le processus de suivi. L'une des difficultés induites par de tels algorithmes porte sur l'efficacité et la précision avec lesquelles les primitives locales de l'image sont combinées avec le modèle global de l'objet. Pour considérer ceci, nous présentons un algorithme de suivi 3D. La pose, c'est-à-dire la position de l'objet dans le repère de la caméra, est calculée pour l'image acquise. Le calcul de pose est un problème de recalage 2D-3D qui consiste à recaler des coordonnées de points 3D (ou autres primitives 3D paramétriques: lignes droites, cercles,...) et leurs projections 2D sur le plan image.

Dans le passé, de nombreux algorithmes d'estimation de pose ont été proposés. Les primitives géométriques considérées pour l'estimation sont généralement des points [9,

4], des segments [6], des contours ou des points sur les contours [16, 18, 7], des coniques [20, 3], des objets cylindriques [5] ou toute combinaison de ces différentes primitives [19]. Une autre question importante est le problème du recalage. Les approches *purement géométriques* [6] ou *itératives* [4] peuvent être considérées. Les *approches linéaires* utilisent des méthodes de type moindres carrés pour estimer la pose. Les techniques *d'optimisation non linéaire* [16, 17, 7, 14] consistent à minimiser l'erreur entre les observations et la projection inverse du modèle. Dans ce cas, la minimisation est réalisée par des algorithmes itératifs numériques tel que Newton-Raphson ou Levenberg-Marquartd, voire minimisation d'hyper-plan [14]. L'avantage principal de ces approches est leur précision. L'inconvénient principal réside dans le risque de tomber dans des minima locaux, voir pire, de diverger. Elles exigent souvent une bonne estimation initiale de la solution pour assurer une convergence correcte. Cependant, dans une application de suivi 3D, le risque de divergence est quasi inexistant, puisque le déplacement de la caméra entre deux images consécutives est relativement petit. Un autre inconvénient est que ces approches ont besoin de l'estimation ou du calcul explicite d'un Jacobien. Une dérivation analytique de ce Jacobien peut être une tâche complexe, tandis que son estimation en ligne peut mener à une minimisation moins efficace et plus côuteuse en temps de calcul.

Dans ce papier une formulation du calcul de pose est proposée laquelle implémente une optimisation non-linéaire : l'Asservissement Visuel Virtuel (AVV). Le calcul de pose est considéré comme un problème d'asservissement visuel 2D [21]. L'asservissement visuel [13, 8, 10] permet de contrôler une caméra par rapport à son environnement. Plus précisément, il consiste à spécifier une tâche (principalement des tâches de positionnement ou des tâches de poursuite de cible) comme la régularisation dans l'image d'un ensemble de primitives visuelles. Un ensemble de contraintes est défini dans le plan image. Une loi de commande, qui minimise l'erreur entre la position actuelle et désirée de ces primitives visuelles, peut être construite automatiquement. L'expérience montre que cette approche est une solution efficace pour des tâches de positionnement d'une caméra dans un contexte robotique (voir les articles dans [10]).

Les avantages principaux d'un asservissement visuel virtuel par rapport aux autres méthodes de calcul de pose non linéaires ou de suivi 3D, peuvent être résumés par les points suivants :

- la formulation proposée facilite la détermination de la forme analytique du Jacobien (appelé matrice d'interaction dans le domaine de l'asservissement visuel) qui est disponible pour de nombreuses primitives visuelles ;
- une nouvelle loi de commande robuste, basée sur un M-Estimateur, est proposée. Elle permet de rejeter des informations visuelles aberrantes;
- différentes lois de commandes peuvent être utilisées et une *analyse de stabilité* peut être effectuée ;
- le suivi se fait à la cadence vidéo ;
- finalement en considérant le calcul de pose comme un problème d'asservissement visuel basé sur l'image, nous profitons de toute la connaissance existante et des résultats dans ce domaine de recherche.

La formulation choisie du suivi dépend entièrement des correspondances entre les primitives dans l'image et le modèle de l'objet. Dans notre méthode, ces correspondances sont données par le suivi local de primitives dans l'image. Plus précisément, l'algorithme des Éléments de Contour en Mouvement [1] est utilisé pour réaliser ce suivi local. Une telle approche locale est sensible aux erreurs de suivi mais elle est, cependant, idéale pour assurer un suivi en temps réel grâce à une recherche 1D le long de la normale aux contours du modèle de l'objet dans l'image. Bien que quelques primitives puissent être incorrectement suivies, l'efficacité globale sera maintenue grâce à l'estimation robuste qui est introduite dans la loi de commande de l'asservissement visuel virtuel.

Kumar [15] donne un synopsis des techniques d'estimation robuste appliqué à la pose. Plus récemment, une approche robuste pour le calcul de pose basée sur l'algèbre de Lie a été proposée [7]. Cette méthode utilise aussi une recherche 1D le long de la normale aux contours et des M-Estimateurs. Cependant, seuls des objets polyédriques ont été considérés, et la forme analytique du Jacobien n'a pas été donnée. Ceci entraîne des difficultés pour l'analyse de la stabilité et du comportement du système. De plus, l'orientation des contours n'est pas considérée pour le suivi local. Ceci dégrade la performance du système en termes de précision des mesures initiales et par conséquent l'efficacité du calcul de pose. Dans notre cas, nous donnons la dérivation complète des matrices d'interaction de type distance à ellipse, droites et cylindres. Précisons qu'il existe une mèthode systématique pour calculer ces matrices d'interaction proposée dans [8]. De plus nous pouvons traiter les primitives en empilant les différents matrices d'interaction associées à chaque primitive. Enfin, notre méthod prendre en compte l'orientation du contour pour obtenir un meilleur suivi local des primitives. Un méthodologie général pour formulé cette matrice et obtenu en prennent avantage du dualité des méthodes d'asservissement visuel.

Dans la suite de ce papier, nous présentons dans la Section 2.1 le principe de l'approche. Dans la Section 2.2, nous exposons les détails des lois de commande de l'asservissement visuel robuste et nous présentons une analyse de la stabilité. Dans la Section 2.3, nous introduisons le calcul de confiance dans l'extraction des primitives locales. Dans la Section 3, nous modélisons les primitives visuelles qui ont été utilisées dans le suivi. Nous dérivons la formulation analytique des matrices d'interaction pour plusieurs primitives. Ensuite nous présentons l'algorithme utilisé pour le suivi local des primitives. Dans la Section 4, nous présentons plusieurs résultats expérimentaux y compris des expériences d'asservissement visuel.

2 Asservissement Visuel Robuste

2.1 Principe

Le principe fondamental de l'approche proposée consiste à définir le problème du calcul de pose comme un problème d'asservissement visuel 2D [8, 13]. L'objectif de l'asservissement visuel est de déplacer une caméra pour observer un objet à une position donnée dans l'image. Ceci est accompli en minimisant l'erreur entre un état désiré des primitives dans l'image ($\mathbf{s_d}$) et leur état courant ($\mp$). Si le vecteur des primitives visuelles est bien choisi, une seule position finale de la caméra permet d'accomplir cette minimisation. Le problème du calcul de pose est semblable.

Pour illustrer le principe, considérons le cas d'un objet avec plusieurs primitives 3D $\mathbf{P}$ (On note ${}^o\mathbf{P}$ les coordonnées 3D de celles-ci dans un repère lié à l'objet). Nous définissons aussi une caméra virtuelle dont la position dans le repère de l'objet est définie par le vecteur $\mathbf{r}$. Le but du problème du calcul de pose est d'estimer les paramètres extrinsèques en minimisant l'erreur Δ entre les données observées $\mathbf{s_d}$ (habituellement la position d'un ensemble de primitives dans l'image) et la position $\mathbf{s}$ des mêmes primitives calculées par une réprojection en accord avec les paramètres extrinsèques et intrinsèques courants :

$$\Delta = \big(\mathbf{s}(\mathbf{r}) - \mathbf{s_d}\big) = \big[\mathbf{pr}_\xi(\mathbf{r},{}^\mathbf{o}\mathbf{P}) - \mathbf{s_d}\big], \qquad (1)$$

où $pr_\xi(\mathbf{r},\cdot)$ est le modèle de la projection d'après les paramètres intrinsèques ξ et la pose de la caméra $\mathbf{r}$. Nous supposons ici que les paramètres intrinsèques ξ sont disponibles mais ils est également possible d'estimer ces paramètres en utilisant la même approche.

Dans la formulation du problème d'asservissement visuel virtuel, nous déplaçons une caméra virtuelle (initialement avec une pose $\mathbf{r_i}$) en utilisant une loi de commande d'asservissement visuel classique avec l'objectif de minimiser cette erreur Δ pour que la caméra virtuelle atteigne la position $\mathbf{r_d}$ qui minimise cette erreur ($\mathbf{r_d}$ correspond à la pose). L'hypothèse que $\mathbf{s_d}$ soit calculé (à partir de l'image) avec une précision suffisante est une supposition importante. En asservissement visuel, la loi de commande qui réalise la minimisation de Δ est traitée habituellement par une approche au moindres carrés [8, 13]. Cependant, si il y a des données aberrantes, une estimation robuste est nécessaire. Les M-estimateurs peuvent être considérés comme une formulation plus générale d'un estimateur au maximum de vraisemblance [12]. Ils sont plus généraux car ils permettent l'utilisation de différentes fonctions de minimisation qui ne correspondent pas nécessairement à une distribution normale des données. Beaucoup de fonctions ont été proposées dans la littérature qui permettent de considérer comme peu vraisemblable des mesures incértaines et dans cértains cas de les rejeter complètement. La fonction a objectif à minimiser est alors modifiée afin de réduire la sensibilité aux données aberrantes. L'équation d'optimisation robuste est donnée par :

$$\Delta_\mathcal{R} = \rho\Big(\mathbf{s}(\mathbf{r}) - \mathbf{s_d}\Big), \qquad (2)$$

où $\rho(u)$ est une fonction robuste [12]. La méthode des Moindres Carrés Pondérés Itérés est une méthode bien connue pour l'utilisation du M-Estimateur. Elle convertit le problème de M-Estimation en un problème équivalent à celui des moindres carrés pondérés.

Afin de combiner une minimisation robuste avec un asservissement visuel, une modification de la loi de commande est nécessaire pour que l'algorithme puisse rejeter des données aberrantes. Nous introduisons dans la loi de commande une matrice de poids diagonale, où les poids reflètent la confiance dans chaque primitive. La nouvelle loi de commande est donnée dans la section suivante et le calcul des poids sera présenté dans la section 2.3.

2.2 Loi de Commande Robuste

L'objectif est de minimiser l'erreur $\|\mathbf{s} - \mathbf{s_d}\|$. En asservissement visuel classique, une fonction de tâche $\mathbf{e}$ est définie par la relation:

$$\mathbf{e} = \mathbf{C}\left(\mathbf{s}(\mathbf{r}) - \mathbf{s_d}\right), \qquad (3)$$

où la matrice $\mathbf{C}$ est appelée la matrice de combinaison. Elle permet de prendre en considération plus de primitives visuelles que le nombre de degrés de liberté contrôlés (6 dans ce cas).

Nous proposons ici, une nouvelle fonction de tâche $\mathbf{e}$ qui assure une minimisation robuste de Δ. Elle est définie par la relation :

$$\mathbf{e} = \mathbf{CD}\left(\mathbf{s}(\mathbf{r}) - \mathbf{s_d}\right), \qquad (4)$$

où $\mathbf{D}$ est une matrice diagonale donnée par

$$\mathbf{D} = \begin{pmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_n \end{pmatrix}$$

et les poids w_i sont donnés par l'équation (14).

En supposant $\mathbf{C}$ et $\mathbf{D}$ constants, la dérivation de l'équation (4) conduit à :

$$\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{s}}\frac{\partial \mathbf{s}}{\partial \mathbf{r}}\frac{d\mathbf{r}}{dt} = \mathbf{CDL_s T_c}, \qquad (5)$$

où $\mathbf{L_s}$ s'appelle la matrice d'interaction [8, 13] associée à $\mathbf{s}$. Elle lie le mouvement des primitives dans l'image à la vitesse $\mathbf{T_c}$ de la caméra virtuelle ($\dot{\mathbf{s}} = \mathbf{L_s T_c}$).

Si nous définissons une décroissance exponentielle de l'erreur $\mathbf{e}$, nous avons :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e}, \qquad (6)$$

où λ est un coefficient proportionnel qui représente le taux de décroissance. Nous pouvons maintenant dériver la loi de commande. En effet, la combinaison de (6) et (5) nous donne :

$$\mathbf{CDL_s T_c} = -\lambda \mathbf{e}. \qquad (7)$$

La loi de commande est finalement donnée par:

$$\mathbf{T_c} = -\lambda(\mathbf{C}\mathbf{D}\mathbf{L_s})^{-1}\mathbf{e}. \quad (8)$$

Dans la pratique, un modèle $\bar{\mathbf{L}}_\mathbf{s}$ de $\mathbf{L_s}$ et un modèle $\bar{\mathbf{D}}$ de $\mathbf{D}$ sont utilisés, et nous obtenons :

$$\mathbf{T_c} = -\lambda(\mathbf{C}\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s})^{-1}\mathbf{e}. \quad (9)$$

La convergence et la stabilité sont des questions importantes lorsqu'on traite une telle loi de commande. En utilisant (9) dans (5), le comportement du système de boucle fermée est obtenu :

$$\dot{\mathbf{e}} = -\lambda\left(\mathbf{C}\mathbf{D}\mathbf{L_s}\right)\left(\mathbf{C}\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s}\right)^{-1}\mathbf{e}. \quad (10)$$

La condition de positivité

$$\left(\mathbf{C}\mathbf{D}\mathbf{L_s}\right)\left(\mathbf{C}\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s}\right)^{-1} > 0 \quad (11)$$

est ainsi suffisante pour assurer la décroissance de $\|\mathbf{e}\|$. Une fois assurée, elle implique la stabilité asymptotique globale et la convergence du système. Cependant, pour obtenir (11), nous avons supposé que $\mathbf{C}$ et $\mathbf{D}$ étaient constant, ce qui n'est pas le cas. Comme pour toute tèchnique actuelle d'asservissement visuel 2D, la stabilité globale ne peut donc pas être démontrée les différents choix possibles de $\mathbf{C}$, $\bar{\mathbf{D}}$ et $\bar{\mathbf{L}}_\mathbf{s}$.
Dans tous les cas considérés ici, la dimension k du vecteur des primitives visuelles $\mathbf{s}$ est plus grande que 6 (les primitives visuelles choisies sont redondantes). Puisque la matrice de combinaison doit être de dimension $6 \times k$ et de rang 6, le choix le plus simple est de définir $\mathbf{C}$ comme la pseudo inverse $(\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s})^+$ de $\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s}$.
Nous avons ainsi $(\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s})^+\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s} = \mathbf{I}_m$, où $\mathbf{I}_m$ est la matrice identité de taille $m * m$. Nous obtenons donc :

$$\mathbf{T_c} = -\lambda\mathbf{e} = -\lambda(\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s})^+\mathbf{D}\big(\mathbf{s}(\mathbf{r}) - \mathbf{s_d}\big), \quad (12)$$

et la condition (11) peut être écrite sous la forme :

$$(\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s})^+\mathbf{D}\mathbf{L_s} > \mathbf{0}. \quad (13)$$

En utilisant cette condition il est possible de considérer l'analyse de la convergence pour trois formulations du modèle $\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s}$ de la matrice d'interaction [2] :

- $\left[\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s}\right]^+ = \left[\mathbf{D}(\mathbf{s_d})\mathbf{L_s}(\mathbf{s_d},\mathbf{r_d}\right]^+$: la matrice d'interaction et les poids sont calculés seulement une fois avec la valeur finale de la pose et les primitives visuelles. Ce choix est le plus classique en robotique. Il assure la stabilité asymptotique locale du système parce que la condition de positivité est assurée dans le voisinage de la position désirée. Ceci signifie que, si l'erreur $\mathbf{s} - \mathbf{s_d}$ est suffisamment petite, la convergence de $\mathbf{s}$ vers $\mathbf{s_d}$ sera obtenue. Pourtant dans notre cas, si $\mathbf{s_d}$ est connu, $\mathbf{r_d}$ (la pose), est inconnue. Ce choix est ainsi impossible pour des applications de suivi.
- $\left[\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s}\right]^+ = \left[\mathbf{D}(\mathbf{s})\mathbf{L_s}(\mathbf{s},\mathbf{r})\right]^+$, les poids et la matrice d'interaction sont calculés à chaque itération avec la valeur actuelle de la pose et des primitives visuelles. Nous pouvons penser que la stabilité globale est démontrée avec $(\mathbf{D}\mathbf{L_s})^+\mathbf{D}\mathbf{L_s} = \mathbf{I_6} > \mathbf{0}$, quelle que soit la valeur de $\mathbf{s}$. Cependant, dans ce cas la matrice $\mathbf{C}$ n'est pas constante et l'équation (5) devraient ainsi tenir compte de la variation de $\mathbf{C}$. Ceci mène à des calculs inextricables, et , seule la stabilité locale peut être obtenue.
- $\left[\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s}\right]^+ = \left[\mathbf{D}(\mathbf{s_i})\mathbf{L_s}(\mathbf{s_i},\mathbf{r_i})\right]^+$ où $\mathbf{r_i}$ est la pose initiale de la caméra virtuelle et $\mathbf{s_i}$ la valeur initiale des primitives visuelles. Ce choix est intéressant parce que $(\bar{\mathbf{D}}\bar{\mathbf{L}}_\mathbf{s})^+$ est calculée seulement une fois. L'évolution des poids pendant la réalisation de la loi de commande sera prise en considération à travers le calcul de $\mathbf{e}$ par $\mathbf{D}(\mathbf{s}(\mathbf{r}) - \mathbf{s_d})$ (voir (4) et (12)). De nouveau, la condition de positivité (13) sera satisfaite seulement si $\mathbf{s_i} - \mathbf{s_d}$ est petite. Ce dernier choix peut être utilisé dans notre application de suivi du fait que $\mathbf{s_i}$ et $\mathbf{r_i}$ sont disponibles.

Nous avons vu que seule la stabilité locale peut être démontrée. Cela signifie que la convergence n'eest pas forcément assurée si l'erreur $\mathbf{s} - \mathbf{s_d}$ est trop grande. Cependant, dans des applications de suivi, le mouvement entre deux images successives, acquises à la cadence vidéo, est suffisamment petit pour assurer la convergence de la loi de commande. En d'autres termes, la convergence est assurée si les résultats obtenus à partir de l'image précédente sont utilisés pour les valeurs initiales $\mathbf{s_i}$ et $\mathbf{r_i}$. En effet, les expériences ont montrée que la convergence est en général obtenue quand le déplacement de la caméra a une erreur d'orientation inférieure à 30^o sur chaque axe. Ainsi des problèmes potentiels existent seulement pour la toute première image où la valeur initiale pour $\mathbf{r}$ ne doit pas être trop mauvaise.

2.3 Calcul du degré de confiance

Les poids w_i, éléments de la matrice $\mathbf{D}$, reflètent la confiance en chaque primitive et sont souvent donnés par [12] :

$$w_i = \frac{\psi(\delta_i/\sigma)}{\delta_i/\sigma}, \quad (14)$$

où $\psi\Big(\delta_i/\sigma\Big) = \frac{\partial}{\partial \mathbf{r}}\rho\big(\delta_i/\sigma\big)$, ψ est la fonction d'influence et δ_i est le résidu normal donné par $\delta_i = \Delta_i - Med(\Delta)$ ($Med(\Delta)$ correspond à la valeur médiane des résidus).
Parmi les diverses fonctions d'influence qui existent dans la littérature, nous avons la fonction de Tukey. Celle-ci rejette complètement les données aberrantes et leur donne un poids nul. En effet, dans cette application de suivi, il est souhaitable que les données aberrantes n'aient aucun effet sur le mouvement de la caméra virtuelle. Cette fonction d'influence est donnée par :

$$\psi(u) = \begin{cases} u(C^2 - u^2)^2 & |u| \leq C \\ 0 & \text{sinon,} \end{cases} \quad (15)$$

où le facteur de proportionnalité pour la fonction de Tukey est $C = 4{,}6851$. Ce facteur représente une efficacité de 95% dans le cas du bruit Gaussien.
Afin d'obtenir une fonction robuste, il est nécessaire de définir un degré de confiance des mesures. L'échelle σ est une évaluation robuste de l'écart type des « bonnes » mesures et est au cœur de la robustesse de la fonction. En utilisant une régression non linéaire pour le calcul de pose, ce facteur d'échelle peut varier enormément au cours de la convergence. C'est souvent traitée comme une variable d'ajustement qui peut être choisie manuellement pour une application particulière. D'autres approches utilisent une statistique robuste pour le calculer. Dans ce article, nous avons retenu l'écart absolu médian (MAD). Il est donné par :

$$\widehat{\sigma} = \frac{1}{\Phi^{-1}(0.75)} Med_i(|\delta_i - Med_j(\delta_j)|), \qquad (16)$$

où $\frac{1}{\Phi^{-1}(0.75)} = 1{,}48$ représente un écart type de la distribution normale et où $\Phi(\cdot)$ est la fonction de distribution cumulée dans le cas Gaussien.
Malheureusement, une preuve de la convergence de la régression non linéaire ne peut être obtenue que si l'on calcul le MAD une seule fois. Ceci est dû à la propriété non asymptotique de la médiane [11]. Par contre, les expérimentations prouvent que le calcul du MAD à chaque itération donne des meilleurs résultats (voir section 4).

3 Primitives visuelles

3.1 Matrices d'interaction

N'importe quel type de primitive géométrique peut être considéré dans la loi de commande proposée dès lors que nous pouvons calculer la matrice d'interaction associée à $\mathbf{L_s}$. Dans [8], un cadre général est proposé pour calculer $\mathbf{L_s}$. C'est un des avantages de cette approche par rapport à d'autres approches non linéaires de calcul de pose. En effet nous pouvons calculer la pose à partir d'un grand nombre d'informations visuelles différents (points, lignes, cercles, quadriques, distances, etc.....). Il est également très facile de mélanger différentes primitives en ajoutant des primitives au vecteur $\mathbf{s}$ et en empilant les matrices d'interaction correspondantes. En outre, si le nombre ou la nature des primitives visuelles sont modifiés avec le temps, la matrice d'interaction $\mathbf{L_s}$ et le vecteur d'erreur $\mathbf{s}$ peuvent être modifiés en conséquence. Dans [18] nous avons appliqué cette approche en considérant les primitives visuelles classiquement utilisées en Asservissement Visuel. Dans cet article nous considérons comme primitive visuelle $\mathbf{s}$ un ensemble de distances entre les primitives locales de type point obtenues à partir d'un processus de traitement d'image et les contours plus globaux d'un modèle. Dans le cas d'une primitive de type distance, $\mathbf{s_d}$ est égal à zéro. Nous avons fait l'hypothèse que les contours de l'objet peuvent être divisés en segments ou en parties linéaires d'ellipses. Tous les points qui correspondent à un segment particulier où une ellipse sont alors traités indépendemment.

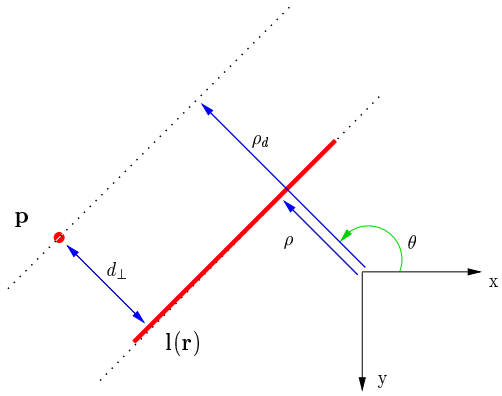

FIG. 1 – *Cas d'une distance à une droite.*

Cas d'une distance à une droite. Nous donnons ici la dérivation de la matrice d'interaction qui lie la variation de la distance entre un point extrait de l'image et une droite virtuelle dont le mouvement est du à la variation de position de la caméra virtuelle. Dans la figure (1), $\mathbf{p}$ est la position du point on extrait et $\mathbf{l}(\mathbf{r})$ est la position de la droite.
La position de la droite est donnée par sa représentation en coordonnées polaire :

$$x\cos\theta + y\sin\theta = \rho, \forall (x,y) \in \mathbf{l}(\mathbf{r}). \qquad (17)$$

La distance entre $\mathbf{p}$ et $\mathbf{l}(\mathbf{r})$ peut être caractérisée par la distance perpendiculaire $d_\perp$ à la droite. La distance parallèle au segment ne contient, elle, aucune information utile à moins qu'une correspondance existe entre un point sur la ligne et $\mathbf{p}$ (ce qui n'est pas le cas ici). Nous avons donc :

$$d_l = d_\perp(\mathbf{p},\mathbf{l}(\mathbf{r})) = \rho(\mathbf{l}(\mathbf{r})) - \rho_\mathbf{d}, \qquad (18)$$

et

$$\rho_d = x_d\cos\theta + y_d\sin\theta, \qquad (19)$$

où x_d et y_d sont les coordonnées du point. On obtient :

$$\dot{d}_l = \dot{\rho} - \dot{\rho_d} = \dot{\rho} + \alpha\dot{\theta}, \qquad (20)$$

où $\alpha = x_d\sin\theta - y_d\cos\theta$. A partir de (20), nous déduisons $\mathbf{L}_{d_l} = \mathbf{L}_\rho + \alpha\mathbf{L}_\theta$. La matrice d'interaction liée à $\mathbf{d_l}$ peut donc être dérivée de la matrice d'interaction liée à une droite (voir [8]) :

$$\begin{aligned}\mathbf{L}_\theta &= \begin{pmatrix} \lambda_\theta\cos\theta & \lambda_\theta\sin\theta & -\lambda_\theta\rho & \rho\cos\theta & -\rho\sin\theta & -1 \end{pmatrix} \\ \mathbf{L}_\rho &= \begin{pmatrix} \lambda_\rho\cos\theta & \lambda_\rho\sin\theta & -\lambda_\rho\rho & (1+\rho^2)\sin\theta & -(1+\rho^2)\cos\theta & 0 \end{pmatrix}\end{aligned} \qquad (21)$$

où $\lambda_\theta = (A_2\sin\theta - B_2\cos\theta)/D_2$ et $\lambda_\rho = (A_2\rho\cos\theta + B_2\rho\sin\theta + C_2)/D_2$. $A_2x + b_2y + c_2z + d_2 = 0$ est l'équation d'un plan 3D lequel appartient à la droite.

Nous obtenons finalement :

$$\mathbf{L_{d_l}} = \begin{pmatrix} \lambda_{d_l}\cos\theta \\ \lambda_{d_l}\sin\theta \\ \lambda_{d_l}\rho \\ (1+\rho^2)\sin\theta - \alpha\rho\cos\theta \\ -(1+\rho^2)\cos\theta - \alpha\rho\sin\theta \\ -\alpha \end{pmatrix}^T, \qquad (22)$$

où $\lambda_{d_l} = \lambda_\rho + \alpha\lambda_\theta$. Notons que le cas de la distance entre un point et la projection d'un cylindre est très semblable.

Cas d'une distance à une ellipse. Nous donnons maintenant la dérivation de la matrice d'interaction qui lie la distance entre un point $\mathbf{p}$ et une ellipse. Cette ellipse correspond à la projection d'un cercle ou d'une sphère en mouvement dans le plan image. Si l'ellipse est paramétrée par son centre de gravité et par ses moments d'ordre 2 (c'est-à-dire $(x_g,y_g,\mu_{02},\mu_{20},\mu_{11})$), la distance $\mathbf{d_e}$ entre le point (x,y) et l'ellipse est définie par :

$$\begin{aligned} d_e = \; & \mu_{02}x^2 + \mu_{20}y^2 - 2\mu_{11}xy + 2(\mu_{11}y_g - \mu_{02}x_g)x \\ & +2(\mu_{11}x_g - \mu_{20}y_g)y + \mu_{02}x_g^2 + \mu_{20}y_g^2 \\ & -2\mu_{11}x_gy_g + \mu_{11}^2 - \mu_{20}\mu_{02}. \end{aligned} \qquad (23)$$

La variation de la distance due à la variation des paramètres d'ellipse est ainsi donnée par :

$$\begin{aligned} \dot{d}_e &= \underbrace{\begin{pmatrix} 2(\mu_{11}(y-y_g)+\mu_{02}(x_g-x)) \\ 2(\mu_{20}(y_g-y)+\mu_{11}(x-x_g)) \\ ((y-y_g)^2-\mu_{02}) \\ 2(y_g(x+x_g)+x_gy+\mu_{11}) \\ ((x-x_g)^2-\mu_{20}) \end{pmatrix}^T}_{\mathbf{L}_e} \begin{pmatrix} \dot{x_g} \\ \dot{y_g} \\ \dot{\mu_{20}} \\ \dot{\mu_{11}} \\ \dot{\mu_{02}} \end{pmatrix} \\ &= \mathbf{L}_e\mathbf{L_c}\mathbf{T_c}, \end{aligned} \qquad (24)$$

où $\mathbf{L_c}$, l'interaction liée à une ellipse, est donnée dans [8].

3.2 Suivi de primitives visuelles

Considérons maintenant le problème de l'extraction d'information visuelles dans les séquences d'images. Pour cela, les déplacements orthogonaux à la projection du modèle sont evalués en utilisant l'algorithme des éléments de contours en mouvement (ECM) [1]. Un des avantages de la méthode des ECM est qu'elle n'exige aucune extraction des contours. Elle récquier seulement la manipulation de coordonnées des points et de leur intensité dans l'image. L'algorithme des ECM peut être mis en oeuvre efficacement puisque seules des convolutions sont utilisées, ceci mène à un calcul en temps réel [1, 18]. Le processus consiste à rechercher le "correspondant" p^{t+1} dans l'image I^{t+1} de chaque point p^t. Nous déterminons un intervalle 1D de recherche $\{Q_j,j \in [-J,J]\}$ dans la direction δ à la normale du contour (voir Figure 2). Pour chaque point p^t et pour chaque position Q_j, se trouvant dans la direction δ^*, nous calculons un critère correspondant à la racine carrée du rapport de log-vraisemblance ζ_j (δ^* est la direction la plus proche de δ dans l'ensemble $\{0^o,45^o,90^o,135^o\}$). Ce rapport, peut se calculer à l'aide de la valeur absolue de la somme des valeurs de convolution, calculée en p^t et Q_j, en utilisant *un masque prédéterminé* M_δ, lequel est fonction de l'orientation du contour.
La nouvelle position p^{t+1} est donnée par :

$$Q^{j^*} = \arg\max_{j\in[-J,J]} \zeta_j \text{ avec } \zeta_j = \mid I^t_{\nu(p^t)} * M_\delta + I^{t+1}_{\nu(Q_j)} * M_\delta \mid$$

tel que ζ_{j^*} est plus grand qu'un seuil λ, et où $\nu(.)$ est le voisinage du pixel considéré.

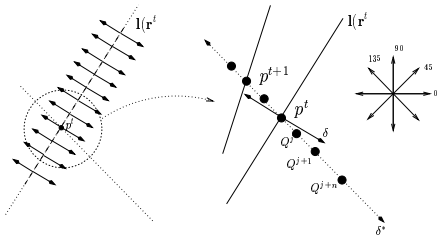

FIG. 2 – *Détermination de la position des points dans l'image suivante avec l'algorithme des ECM.*

À l'issue de cette étape, nous disposons à une liste de k pixels ainsi que de leur distance $d_\perp$ à la ligne de support (ou d_e pour une ellipse). Ce processus est exécuté pour chaque nouvelle image et n'exige jamais l'extraction de nouveaux contours.

4 Résultats expérimentaux

Dans les quatre expériences dessous, des séquences d'images "réelles" ont été acquises en utilisant un caméscope. Dans de telles expériences, le traitement d'image est potentiellement très complexe. En effet l'extraction et le suivi de points fiables dans un environnement réel est souvent difficile. Nous démontrons l'utilisation de primitives telles que la distance à la projection des cercles, droités, et cylindres 3D. Dans toutes les expériences, les distances sont calculées en utilisant l'algorithme des ECM décrit précédemment. Le suivi est toujours exécuté à une cadence compatible avec la cadence vidéo. Dans la plupart des expériences une recherche de 6 pixels à la normale du contour est efféctuée avec une masque de taille 10 et nous efféctuons cette recherche avec un échantillonage de 6 pixels sur les contours de la projection du modèle. Ces paramètres peuvent varier selon la scène et la complexité du modèle afin de conserver la cadence vidéo. Par exemple, avec des contours bien contrastés, il n y a pas besoin d'un masque de taille aussi grande et on peut effectuer un recherche de plus grande amplitude. Un gain $\lambda = 0.7$ été utilisé pour ces expérimentations.

Suivi dans un environnement d'intérieur. Dans la première expérience, nous montrons le résultat du suivi de quatre cercles 3D (voir Figure 3. Cette très longue séquence (plus de 2000 images) contient des occultations multiples de cértains de ces cercles. Dans cette expérience, bien que les images soient simples, si aucune évaluation robuste de

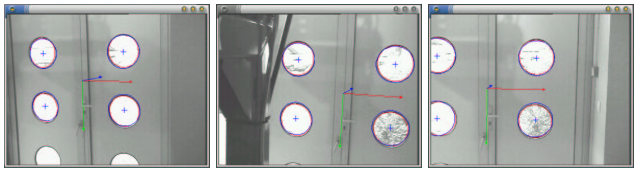

FIG. 3 – *Suivi de cercles 3D. Quatre cercles sont suivis le long d'une séquence de 2000 images. Cette séquence comporte des occultations multiples de cértains cercles.*

la pose à l'aide des M-estimateurs n'est considérée, le processus de minimisation doit traiter trop de mauvais appariements dus aux occultations, et le suivi échoue après quelques images.
Dans la deuxième expérience (voir Figure 4, un objet dont le modèle est fait d'un cylindre, un cercle et deux lignes est considéré. Ceci illustre la possibilité qu'a notre algorithme de considérer des primitives diverses dans le même processus de minimisation.

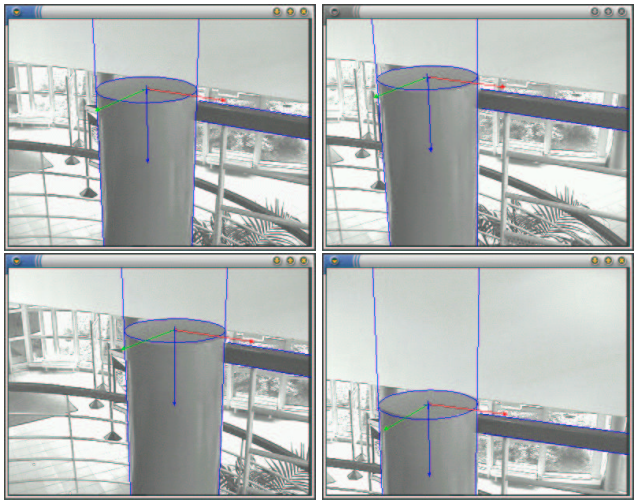

FIG. 4 – *Suivi simultané d'un cercle, d'un cylindre et de deux lignes droites.*

Suivi dans un environnement d'extérieur. Dans la troisième expérience (voir Figure 5), une scène d'extérieur est considérée. Ici, les distances à la projection d'un cylindre 3D et à deux lignes 3D sont employées pour calculer la pose. Malgré des images très bruitées (vent dans les arbres, occultations multiples, etc...) le suivi est réalisé le long d'une séquence de 1400 images. Les images montrent les lignes et les limbes du cylindre suivi (en rouge) ainsi que des informations 3D (en bleu) insérées après le calcul de pose (les repères de référence et la projection du cylindre et des lignes). La figure 6 montre comment cet algorithme de suivi 3D peut être utilisé dans une application de realité augmentée.

Suivi pour l'asservissement visuel. Le suivi d'objet en temps-réel est souvent considéré comme un frein au devéloppement des techniques d'asservissement visuel. Cette

FIG. 5 – *Suivi dans un environnement d'extérieur. Malgré des occultations multiples et des perturbations, le suivi est toujours très fiable et est géré en temps réel.*

FIG. 6 – *Suivi en utilisant un cylindre et deux lignes droites. Une application à la réalité augmentée.*

application exige un algorithme de suivi à la fois rapide et fiable. Nous avons considéré ici une tâche de positionnement. A partir d'une première position, le robot doit atteindre une position de l'espace définie par une position désirée de l'objet dans l'image. L'exécution complète de cette tâche d'asservissement visuel (le suivi et la commande) ont été realisé sur une plate-forme expérimentale composé d'une caméra CCD montée sur l'effecteur d'un robot à six degrés de liberté. L'objet d'intérêt pour cette expérience était un micro contrôleur composé de plusieurs primitives de type droites.
Pour valider la robustesse de l'algorithme, le microcontrôleur a été placé dans un environnement fortement texturé (voir Fig. 7). Les tâches de suivi et de positionnement ont été correctement réalisées. Les images ont été acquises et traitées à une cadence de 25Hz. Plusieurs occultations partielles (main, outils, etc) ainsi que des variations d'éclairage importants ont été provoquées pendant la réalisation de la tâche de positionnement.

5 Conclusion

Pour des applications de suivi, le calcul de la pose de la caméra est utile. Cet article s'est focalisé sur la détermination de la pose d'objet dans des séquences d'image. Cette

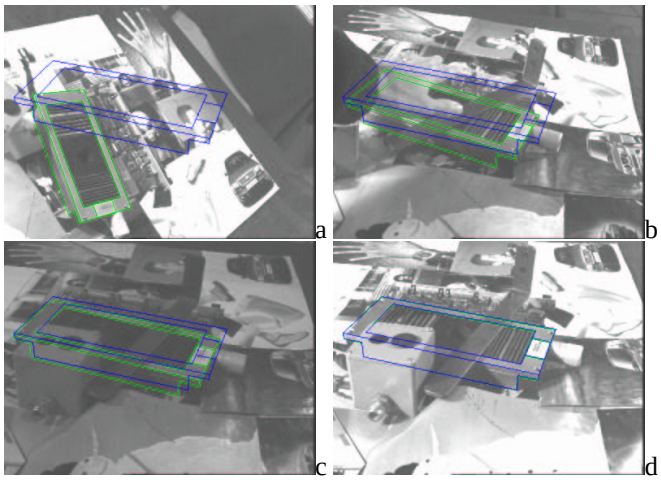

FIG. 7 – *Expériences d'asservissement visuel: Dans ces images on peut voir que l'algorithme est très robuste aux occultations partielles et aux variations importantes de l'éclairage. Les images sont acquises et traitées à la cadence de trame (25Hz). La position désirée à atteindre est en bleu. La position estimeé après chaque calcul de pose et en vert. (a) image initiale (b) occultation partielle de l'objet (c) variation d'éclairage (d) image finale avec diverses occultations*

pose est déterminée par une technique d'asservissement visuel virtuel. Les matrices d'interaction, lesquelles lient la vitesse de la caméra virtuelle à la variation des primitives visuelles dans l'image, ont été déterminées. Elle agissent comme le Jacobien dans une approche de minimisation non linéaire classique. Dans cet article des primitives de type distance ont été determinées afin de représenter un lien entre des points 2D et la projection des lignes droites, cercles et cylindres 3D. Une nouvelle loi de commande robuste a été proposée intégrant des M-estimateurs. L'algorithme de calcul de pose peut ainsi traiter efficacement les erreurs de suivi qui contribuent habituellement à une dégradation du système jusqu'à un arrêt définitif. Les résultats expérimentaux présentés ont été obtenus en utilisant plusieurs caméras, objectifs, et environnements. L'algorithme a été testé avec des séquences d'images diverses et des applications diverses (asservissement visuel, réalité augmentée...) ce qui démontre l'intérêt de notre approche. A chaque fois le suivi a été implémenté en temps réel. Des travaux futurs seront consacrés au cas des objets déformables et à la reconstruction de modèles paramétriques de tels objets.

Références

[1] P. Bouthemy. A maximum likelihood framework for determining moving edges. *IEEE Trans. on Pattern Analysis and Machine intelligence*, 11(5):499–511, May 1989.

[2] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G . Hager, et A.S. Morse, editors, *The Confluence of Vision and Control*, pages 66–78. LNCIS Series, No 237, Springer-Verlag, 1998.

[3] S. de Ma. Conics-based stereo, motion estimation and pose determination. *Int. Journal of Computer Vision*, 10(1):7–25, 1993.

[4] D. Dementhon et L. Davis. Model-based object pose in 25 lines of codes. *Int. J. of Computer Vision*, 15:123–141, 1995.

[5] M. Dhome, J.-T. Lapresté, G. Rives, et M. Richetin. Determination of the attitude of modelled objects of revolution in monocular perspective vision. In *European Conference on Computer Vision, ECCV'90*, volume LNCS 427, pages 475–485, Antibes, April 1990.

[6] M. Dhome, M. Richetin, J.-T. Lapresté, et G. Rives. Determination of the attitude of 3-d objects from a single perspective view. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, December 1989.

[7] T. Drummond et R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(7):932–946, July 2002.

[8] B. Espiau, F. Chaumette, et P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[9] R. Haralick, H. Joo, C. Lee, X. Zhuang, V Vaidya, et M. Kim. Pose estimation from corresponding point data. *IEEE Trans on Systems, Man and Cybernetics*, 19(6):1426–1445, November 1989.

[10] K. Hashimoto (eds). *Visual Servoing : Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific Series in Robotics and Automated Systems, Vol 7, World Scientific Press, Singapor, 1993.

[11] P.-W. Holland et R.-E. Welsch. Robust regression using iteratively reweighted least-squares. *Comm. Statist. Theory Methods*, A6:813–827, 1977.

[12] P.-J. Huber. *Robust Statistics*. Wiler, New York, 1981.

[13] S. Hutchinson, G. Hager, et P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.

[14] F. Jurie et M. Dhome. Read time 3d template matching. *International Conference on Computer Vision and Pattern Recognition*, 1:791, December 2001.

[15] R. Kumar et A.R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *CVGIP: Image Understanding*, 60(3):313–342, Novembre 1994.

[16] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–394, 1987.

[17] C.P. Lu, G.D. Hager, et E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE trans on Pattern Analysis and Machine Intelligence*, 22(6):610–622, June 2000.

[18] E. Marchand, P. Bouthemy, F. Chaumette, et V. Moreau. Robust real-time visual tracking using a 2d-3d model-based approach. In *IEEE Int. Conf. on Computer Vision, ICCV'99*, volume 1, pages 262–268, Kerkira, Greece, September 1999.

[19] E. Marchand et F. Chaumette. Virtual visual servoing: a framework for real-time augmented reality. In *EUROGRAPHICS'02 Conference Proceeding*, volume 21(3) of *Computer Graphics Forum*, pages 289–298, Saarebrücken, Germany, September 2002.

[20] R. Safaee-Rad, I. Tchoukanov, B. Benhabib, et K.C. Smith. Three dimentional location estimation of circular features for machine vision. *IEEE trans on Robotics and Automation*, 8(2):624–639, october 1992.

[21] V. Sundareswaran et R. Behringer. Visual servoing-based augmented reality. In *IEEE Int. Workshop on Augmented Reality*, San Francisco, November 1998.