



**HAL**  
open science

## Suivi 3D temps réel : approche contour/texture

M. Pressigout

► **To cite this version:**

M. Pressigout. Suivi 3D temps réel : approche contour/texture. MajecSTIC 2005 : Manifestation des Jeunes Chercheurs francophones dans les domaines des STIC, IRISA – IETR – LTSI, Nov 2005, Rennes, France. pp.57-64. inria-00000703

**HAL Id: inria-00000703**

**<https://inria.hal.science/inria-00000703>**

Submitted on 15 Nov 2005

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Suivi 3D temps-réel: approche contour/texture

Muriel Pressigout  
Projet Lagadic  
Université de Rennes I, IRISA/INRIA  
Campus de Beaulieu, 35000 Rennes  
Muriel.Pressigout@irisa.fr

## Abstract

*Cet article présente une méthode de suivi 3D basé modèle, précise et robuste. L'information apportée par le motif de l'objet est intégrée à un algorithme classique de calcul de pose reposant sur ses contours de façon à obtenir un résultat plus fiable. La robustesse du suivi est assurée par l'utilisation des M-estimateurs dans le processus de calcul. L'approche a été validée sur des séquences vidéos ainsi que lors d'applications " en live " de réalité augmentée ou en robotique. Ces résultats montrent que la méthode proposée est robuste aux larges mouvements et à des environnements réalistes.*

*This paper proposes a real-time, robust and efficient 3D model-based tracking algorithm. A virtual visual servoing approach is used for monocular 3D tracking. The integration of texture information in the classical non-linear edge-based pose computation provides a more reliable tracker. Robustness is enforced by integrating a M-estimator into the virtual visual control law via an iteratively re-weighted least squares implementation. The method presented in this paper has been validated on several video sequences as well as during visual servoing experiments considering various objects. Results show the method to be robust to large motions and textured environments.*

## 1. Introduction

Le problème considéré dans cet article est l'estimation dans l'espace tridimensionnel de la position et de l'orientation d'une caméra par rapport à la scène qu'elle filme en se basant sur les informations visuelles extraites des images acquises. Ce sujet, nommé suivi 3D ou calcul de pose, a donné lieu à de nombreux travaux aux applications diverses et variées. Les applications qui motivent ces travaux de recherche sont le déplacement de robots basé sur l'utilisation d'une caméra et la Réalité Augmentée (RA) où des objets virtuels sont insérés dans un flot d'images. Ceci

contraint à effectuer les calculs en temps-réel (25 Hz) pour obtenir un système réactif. Nous supposons que le modèle 3D de l'objet est connu. Cette approche qui se classe donc dans la famille des algorithmes basés modèle intègre l'information basée sur le motif de l'objet à un processus basé sur l'analyse des contours de l'objet à suivre de façon à obtenir une méthode de suivi fiable et plus robuste que les méthodes classiques.

En vision par ordinateur, la plupart des techniques de suivi disponibles peuvent être divisées en deux classes principales. La première est basée uniquement sur l'analyse des informations extraites dans les images, c'est-à-dire sur des primitives géométriques 2D (points, segments, cercles, ...), les contours de l'objet [11], des régions d'intérêt [7]... La seconde utilise en plus explicitement un modèle des objets suivis qui peut être un modèle 3D DAO [4, 5, 14, 16] ou un modèle 2D de l'objet [13]. Cette catégorie de méthodes fournit habituellement une solution plus robuste. L'avantage principal des méthodes basées modèle est que la connaissance du sujet de la scène (l'information 3D implicite) permet l'amélioration des résultats par la prévision des mouvements cachés de l'objet, la détection des occultations partielles et agit de manière à réduire les effets des données erronées dans le processus de suivi.

L'algorithme de suivi 3D proposé dans cet article rentre dans la dernière catégorie: il s'appuie sur le modèle 3D de l'objet pour compléter l'analyse des primitives 2D extraites des images. Dans un tel cas, le processus de suivi se base habituellement sur une approche mono-image d'estimation de la pose. Classiquement, la solution est donnée par un recalage 2D-3D *ie.* par l'alignement des données 2D extraites des images avec celles obtenues grâce au modèle 3D. Dans la littérature relative à ce problème, les primitives géométriques considérées pour le calcul de pose sont souvent des points [3], des contours ou des points sur le contour [4, 16], des segments, des lignes, des coniques, des objets cylindriques ou une combinaison de ces différents primitives [18]. De telles primitives sont extraites à partir des im-

ages après un processus d'extraction des contours. Utiliser uniquement l'information portée par les contours fournit de bons résultats lorsque les contours sont bien marqués même s'il y a des changements d'illumination. Cependant, cela peut donner un suivi assez irrégulier voire fausser l'estimation de la pose si l'environnement ou l'objet comporte des motifs ambigus.

L'information portée par les motifs de l'objet est utilisée dans beaucoup de travaux pour le suivi d'un objet dans une séquence d'image. Contrairement aux algorithmes basés sur les contours de l'objet, elle est bien adaptée aux objets texturés et ne souffre habituellement pas d'irrégularités dans l'estimation des paramètres. Cependant, cette solution n'est pas appropriée pour les objets peu texturés et est principalement exploitée dans le 2D suivi, tel que l'algorithme KLT [22] ou le suivi de région d'intérêt [6, 12]. Des points d'intérêt peuvent également être employés pour le suivi 3D dans le phase de recalage 2D-3D [19, 23]. Les motifs ont également été exploités dans [21] pour trouver la projection des contours d'un objet 3D. Ceci est réalisé en remplaçant la détection standard basée gradient par une méthode qui calcule l'endroit le plus probable de la frontière de la texture. Dans [12], les valeurs de niveaux de gris sont intégrées directement dans le processus de minimisation du suivi 3D. Une approche basée texture peut souffrir d'un manque de précision si la taille de l'objet dans l'image change suffisamment.

Comme on peut le remarquer, les algorithmes de suivi basés modèle peuvent être principalement classés en deux catégories, suivant qu'ils se basent sur l'analyse des contours de l'objet ou de son motif, traitant chacun différents types d'objets ou d'environnement. Cependant, dans une séquence vidéo visuel réaliste, la différence entre chaque cas n'est pas aussi claire. En outre, les avantages et les inconvénients de chaque catégorie sont complémentaires. L'idée est alors d'intégrer les deux approches dans le même processus. [20] proposent par exemple d'estimer le mouvement apparent de l'objet dans l'image selon ce principe. [17] effectue un suivi 2D basé sur l'estimation du mouvement dominant pour initialiser le suivi 3D basé sur la projection de bord. Fusionner les deux approches pour effectuer un suivi 3D a été étudié dans les travaux récents de- [24, 15]. [24] rassemble dans un filtre de Kalman des mesures sur le centre de gravité de l'objet, sur les couleurs, les orientations et les positions des contours et des déplacements de primitive obtenus par une minimisation SSD des différences entre les niveaux de gris prédits et ceux de l'image courante. Dans [15], l'approche basée modèle proposée considère en même temps le recalage 2D-3D par rapport à des images clés (qui fournit une pose pour chaque image) comme dans une approche modèle-basée classique (par exemple [2, 4]), en considérant des hypothèse multiples pour le suivi des contours, et le recalage temporel

2D-2D (qui impose des contraintes spatio-temporelles entre vues différentes d'une même scène dans le processus de suivi). Le cadre présenté ici fusionne également une approche classique basée sur l'extraction de contour et un recalage temporel basé sur l'analyse des motifs dans une fonction objectif non linéaire à optimiser. En effet, estimer conjointement la pose et le déplacement de la caméra impose une contrainte spatio-temporelle implicite qui manque à un algorithme de suivi basé modèle classique. La fusion est cependant gérée d'une manière différente que dans [15] et ne nécessite pas d'extraction de points d'intérêt dans chaque image ou d'architecture matérielle spécifique.

Il faut noter au passage que de nombreuses méthodes se basent sur un cadre bayésien comme dans [11]. Bien que ce type d'approche soit très intéressant et extensivement utilisé, le suivi bayésien est une technique très différente que celle présentée dans cet article et quoique les objectifs soient semblables, l'aspect théorique est très différent et peut à peine être comparé.

Dans cet article, la pose et le calcul de déplacement de la caméra sont formulés en terme d'optimisation non linéaire en utilisant les techniques d'Asservissement Visuel Virtuel (AVV). Les deux problèmes sont comparés de façon similaire à l'asservissement visuel 2D comme expliqué dans [2]. L'asservissement visuel 2D [10] consiste à contrôler les mouvements d'un robot par l'analyse des informations visuelles fournies par une caméra. La tâche du robot (principalement des tâche de positionnement ou de suivi) est alors spécifiée comme la régulation dans l'image d'un ensemble de primitives visuelles. Une loi de commande en boucle fermée qui réduit au minimum l'erreur entre la position désirée et la position courante de ces primitives visuelles peut alors être mise en application. Elle détermine automatiquement le mouvement que la caméra doit réaliser. Ce principe est utilisé pour créer un système de suivi basé sur les primitive extraites des images capable de traiter des scènes complexes en temps réel. Des avantages de la formulation par asservissement visuel virtuel sont discutés dans [2] (précision, efficacité, stabilité, et problèmes de robustesse). Pour améliorer la robustesse, un M-estimateur est intégré dans une loi robuste de commande. L'algorithme de calcul de pose ou de déplacement résultant peut ainsi traiter efficacement les primitives dont le suivi est erroné sans dégrader le comportement de l'algorithme.

Dans la suite, la Section 2 présente le principe de l'approche en utilisant des techniques d'AVV. La Section 3 décrit les primitives choisies pour effectuer une estimation de la pose ou du déplacement de la caméra. Les détails de l'intégration de l'estimation du déplacement de la caméra dans le processus de calcul de pose sont donnés dans Section 4. Afin de valider cette approche l'algorithme est testé sur plusieurs séquences vidéo réalistes. Ces résultats expérimentaux sont rapportés dans Section 5.

## 2. Suivi basé modèle par minimisation non linéaire

Le principe fondamental de l'approche proposée est d'intégrer une estimation du déplacement de la caméra basée sur l'information donnée par les motifs dans un processus plus classique de calcul de pose de la caméra qui s'appuie sur les primitives basées contour. Ceci est réalisé en utilisant des techniques d'asservissement visuel virtuel (AVV). En effet, les problèmes d'estimation de pose et de déplacement peuvent être définis tous les deux comme le problème dual de l'asservissement visuel 2D [10]. Cette section est consacrée à la description du cadre général du processus d'estimation.

En asservissement visuel, le but est de déplacer une caméra afin d'observer un objet à une position donnée dans l'image. Le problème de calcul de pose ou de déplacement de la caméra est très semblable. Une caméra virtuelle est définie par sa position  $\mathbf{r}$  dans le repère d'objet. L'approche consiste à estimer la vraie pose ou le vrai déplacement en minimisant l'erreur  $\Delta$  entre les données observées  $s^*$  et la valeur courante  $s$  des mêmes primitives calculées par une projection selon la pose ou le déplacement courant :

$$\Delta = \sum_{i=1}^N \rho(s_i(\mathbf{r}) - s_i^*)^2, \quad (1)$$

où  $\rho(u)$  est une fonction robuste [9]. Cette formulation de l'erreur est utilisée pour le calcul de pose de la caméra et le calcul de déplacement, toutefois les primitives  $s$  seront différentes dans chaque cas. En se basant sur cette formulation du problème, une caméra virtuelle initialement à la position  $\mathbf{r}_1$  est déplacée en utilisant une loi de commande calculée par asservissement visuel afin de minimiser l'erreur  $\Delta$ . À la convergence, la caméra virtuelle a réalisé le positionnement ou le déplacement qui minimise l'erreur  $\Delta$ .

Cet objectif est incorporé à une loi de commande robuste. Ainsi, de manière identique à un processus IRLS (Iteratively Reweighted Least Square), l'erreur à réguler à 0 est définie par :

$$\mathbf{e} = \mathbf{D}(s(\mathbf{r}) - s^*), \quad (2)$$

où  $\mathbf{D}$  est une matrice de poids diagonale donnée par  $\mathbf{D} = \text{diag}(w_1, \dots, w_k)$ . Les poids  $w_i$  reflètent la confiance en chaque primitive et leur calcul est basé sur les M-estimateurs [2]. Une loi simple de commande peut alors être conçue pour essayer d'assurer une diminution exponentielle de  $\mathbf{e}$  autour de la position désirée  $s^*$ . La loi de commande est donnée par :

$$\mathbf{v} = -\lambda(\widehat{\mathbf{D}\mathbf{L}_s})^+ \mathbf{D}(s(\mathbf{r}) - s^*), \quad (3)$$

où  $\mathbf{v}$  est le torseur cinématique de la caméra virtuelle et  $\mathbf{L}_s$  la matrice d'interaction liée à  $s$  et définie par  $\dot{s} = \mathbf{L}_s \mathbf{v}$ .

N'importe quel genre de primitives peut être considéré dans la loi de commande proposée dès qu'il est possible de calculer sa matrice d'interaction correspondante  $\mathbf{L}_s$ . En effet, il est possible de calculer la pose de la caméra à partir d'un grand ensemble d'information basée contour (points, lignes, cercles, distances, etc....) ou le déplacement de la caméra à partir de l'information portée par la texture dans un même cadre. La combinaison de différents primitives est réalisée en ajoutant des primitives au vecteur  $s$  et " en empilant " la matrice d'interaction correspondante de chaque primitive dans une grande matrice d'interaction de la taille  $nd \times 6$  où  $n$  correspond au nombre de primitives et  $d$  à leur dimension :

$$\begin{pmatrix} \dot{s}_1 \\ \vdots \\ \dot{s}_n \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{s1} \\ \vdots \\ \mathbf{L}_{sn} \end{pmatrix} \mathbf{v} \quad (4)$$

La redondance donne un résultat plus précis de la pseudo-inverse de  $\mathbf{L}_s$  définie dans (3).

Selon la nature des primitives, les techniques de AVV peuvent résoudre un problème de calcul de pose ou de déplacement de la caméra. La combinaison des deux approches introduit une contrainte spatio-temporelle dans l'estimation de la pose en considérant des informations sur l'objet extraites dans l'image courante et les précédentes ainsi que les contraintes géométriques sous-jacentes. La Section suivante est consacrée au choix des primitives visuelles et à leur rôle dans la méthode proposée.

## 3. Choix des primitives

La première sous-section est consacrée aux primitives basées contour utilisées dans un calcul de pose classique. La suivante présente les primitives basées sur la texture pour l'estimation du déplacement.

### 3.1. Calcul de pose: primitives basées contour

La prise en compte de primitives visuelles basées contour permet d'effectuer un calcul de pose classique [2, 4, 16]. Pour illustrer le principe de cette approche, considérons le cas d'un objet composé de différentes primitives géométriques  ${}^o\mathbf{P}$  (par exemple  ${}^o\mathbf{P}$  représente les coordonnées 3D de ces primitives dans le repère de l'objet). L'idée est d'estimer la pose réelle en minimisant l'erreur  $\Delta$  entre les données observées  $s^*$  (la position d'un ensemble de primitives dans l'image dans le cas du calcul de pose) et la position  $s$  de ces mêmes primitives calculée par une projection selon les paramètres courants de la pose :

$$\Delta = \sum_{i=1}^N \rho(pr_{\xi}(\mathbf{r}, {}^o\mathbf{P}_i) - s_i^*)^2, \quad (5)$$

où  $pr_{\xi}(\mathbf{r}, \circ \mathbf{P})$  est le modèle de projection dépendant des paramètres intrinsèques  $\xi$  et de la pose  $\mathbf{r}$  de la caméra.  $\mathbf{r}$  est un vecteur stockant les six paramètres de pose : deux pour la translation et la rotation sur chacun des trois axes. On suppose ici que les paramètres intrinsèques  $\xi$  sont disponibles mais il est possible, en utilisant la même approche, d'estimer également ces paramètres. À la convergence, la caméra virtuelle atteint la pose  $\mathbf{r}$  qui minimise l'erreur  $\Delta$  ( $\mathbf{r}$  sera la pose réelle de la caméra).

Dans le cas d'un suivi basé contour, le cadre de l'AVV permet de traiter différents genres de primitives géométriques en utilisant (4). La dérivation de la matrice d'interaction pour différents primitives géométriques est décrite dans [2, 4, 16]. Lors de nos travaux, nous avons considéré des primitives correspondant à une distance entre les contours du modèle DAO reprojété dans l'image et des points qui appartiennent aux contours extraits dans l'image courante selon [1]. Dans ce cas-ci la valeur désirée des distances est égale à zéro. En se basant sur l'hypothèse que les contours de l'objet dans l'image peuvent être décrites par morceaux linéaires, toutes les distances sont traitées en fonction du segment associé.

Le suivi basé modèle et basé contour correspond à une méthode classique de calcul de pose. Il est rapide, efficace, robuste aux changements d'illumination. Cependant, c'est principalement un processus mono d'image ce qui implique certains défauts. Si les primitives géométriques ne peuvent pas être exactement extraites sans aucune ambiguïté, le suivi peut manquer de précision. Par conséquent, il est sensible aux motifs de l'objet ou du fond. Puisque le processus se fonde principalement sur l'analyse de l'image courante sans tenir compte du passé du suivi (sauf pour l'initialisation), ceci peut entraîner une divergence dans le suivi.

### 3.2. Estimation du déplacement de la caméra: primitives basées texture

L'idée est alors d'intégrer des informations sur le passé pour exécuter un suivi spatio-temporel afin de corriger les inconvénients du suivi basé contour présenté dans le paragraphe précédent. Ceci est réalisé par une estimation du déplacement de la caméra basée sur la mise en correspondance des intensités des niveaux de gris entre deux images incorporée dans le même cadre que le calcul de pose.

Si la pose  ${}^1\mathbf{M}_o$  de la caméra dans la première image est connue, le calcul du déplacement  ${}^2\mathbf{M}_1$  de la caméra ou de la pose  ${}^2\mathbf{M}_o$  de la caméra sont exactement les mêmes problèmes<sup>1</sup>. Dans les deux cas, la vitesse de la caméra est calculée pour mettre à jour la pose ou le déplacement de la

<sup>1</sup>La matrice  $4 \times 4$   ${}^i\mathbf{M}_o$  correspond au changement de repère dont les six paramètres en translation et rotation sont stockés dans le vecteur  $\mathbf{r}_i$

caméra, ce qui revient au même puisque :

$${}^2\mathbf{M}_o = {}^2\mathbf{M}_1 \cdot {}^1\mathbf{M}_o \quad (6)$$

Alors que pour l'estimation de pose le but est de minimiser l'erreur entre les primitives observées dans l'image et leur projection sur le plan de l'image, pour l'estimation du mouvement de la caméra l'idée est de minimiser l'erreur entre la valeur des niveaux de gris à la position  $\mathbf{p}_1$  dans la première image  $\mathbf{I}_1$  et celles observées dans la deuxième image  $\mathbf{I}_2$  à la position des primitives correspondantes transférées à partir de  $\mathbf{I}_1$  dans  $\mathbf{I}_2$  par une transformation 2D  ${}^2tr_1$ .  ${}^2tr_1$  dépend du déplacement de la caméra et des contraintes géométriques entre plusieurs vues d'une même scène. L'équation (1) est alors :

$$\Delta = \sum_{i=1}^N \rho(I_1(\mathbf{p}_{1_i}) - I_2({}^2tr_1(\mathbf{p}_{1_i})))^2, \quad (7)$$

où  $N$  est le nombre de pixels pris en compte. Les aspects concernant la transformation 2D et la matrice... seront vus par la suite. À la convergence, la caméra virtuelle a réalisé le déplacement  ${}^2\widehat{\mathbf{M}}_1$  qui minimise cette erreur ( ${}^2\widehat{\mathbf{M}}_1$  sera le vrai déplacement de la caméra).

Un tel processus permet l'intégration des primitives basées texture exploitées pour l'estimation de déplacement de la caméra dans la loi de commande utilisée pour le calcul de pose suivant (4).

## 4. Intégrer l'estimation du le déplacement de la caméra dans le processus de calcul de pose

Cette Section présente les détails de l'intégration de l'estimation de déplacement dans le processus de calcul de pose. L'algorithme résultant s'appelle le suivi hybride. Les premiers aspects qui vont être évoqués dans cette Section sont la transformation 2D et la matrice d'interaction associée à (7). Le modèle et les détails sur le traitement des données sont présentés à la fin.

**Structure planaire.** Supposons dans un premier temps que le motif est contenu dans un plan. De ce fait, un point  $\mathbf{p}_1$  dans l'image  $\mathbf{I}_1$  exprimé en coordonnées homogènes  $\mathbf{p}_1 = ({}^1u, {}^1v, 1)$ , est transféré dans l'image  $\mathbf{I}_2$  au point  $\mathbf{p}_2$  par :

$$\mathbf{p}_2 = {}^2tr_1(\mathbf{p}_1) \propto \mathbf{K}^{-1} {}^2\mathbf{H}_1 \mathbf{K} \mathbf{p}_1, \quad (8)$$

où  $\mathbf{K}$  est la matrice des paramètres intrinsèques de la caméra et  ${}^2\mathbf{H}_1$  est une homographie (définie jusqu'à un facteur d'échelle) qui définit la transformation en coordonnées métriques entre les images acquises par la caméra à la pose 1 puis 2. Une fois qu'un déplacement de la caméra est produit, l'homographie  ${}^2\mathbf{H}_1$  est donnée par :

$${}^2\mathbf{H}_1 = ({}^2\mathbf{R}_1 + \frac{{}^2\mathbf{t}_1}{{}^1d} \mathbf{1} \mathbf{n}^\top), \quad (9)$$

où  ${}^1\mathbf{n}$  et  ${}^1d$  sont respectivement la normale du plan et la distance à l'origine du plan de référence exprimées dans le repère de la caméra 1.  ${}^2\mathbf{R}_1$  et  ${}^2\mathbf{t}_1$  sont respectivement la matrice de rotation et le vecteur de translation entre les deux repères de la caméra. Nous obtenons finalement  $\mathbf{p}_2 = {}^2tr_1(\mathbf{p}_1) = ({}^2u, {}^2v, {}^2w)$  qui sera utilisé dans la prochaine itération du processus de minimisation.

**Structure non-planaire.** Dans le cas d'une structure non-planaire, le transfert de point donné par (8) devient [8] :

$$\mathbf{p}_2 = {}^2tr_1(\mathbf{p}_1) = \mathbf{K}^{-1} {}^2\mathbf{H}_1 \mathbf{K} \mathbf{p}_1 + \beta_1 \mathbf{c}_2, \quad (10)$$

où  ${}^2\mathbf{H}_1$  est l'homographie induite par le plan de référence  $\pi$  comme vu dans le paragraphe précédent, le scalaire  $\beta_1$  est la parallaxe relative à l'homographie  ${}^2\mathbf{H}_1$  et  $\mathbf{c}_2 = \mathbf{K}^2 \mathbf{t}_1$  l'épipolette projeté sur l'image 2 en coordonnées pixelliques.  $\beta_1$  peut être interprétée comme une profondeur relative au plan  $\pi$  :

$$\beta_1 = \frac{d_1 - {}^1\mathbf{n}^\top (Z_1 \mathbf{K}^{-1} \mathbf{p}_1)}{Z_1 d_1} \quad (11)$$

avec  $Z_1$  la coordonnée en profondeur du point 3D associée à  $\mathbf{p}_1$  exprimé dans le repère 1 de la caméra.

Puisque que  $\beta_1$  dépend seulement de paramètres exprimés dans le repère de la caméra 1, il peut être précalculé. La valeur de  $Z_1$  est donnée par l'intersection de la structure 3D et du rayon allant du centre de la caméra à  $\mathbf{p}_1$ . Pour une sphère dont le rayon est  $r_s$  et le centre  $\mathbf{c}_{s1} = (x_{s1}, y_{s1}, z_{s1})$ , on a :

$$Z_1 = \frac{b - \sqrt{b^2 - ac}}{a} \quad (12)$$

avec :

$$a = x_1^2 + y_1^2 + 1 \quad (13)$$

$$b = x_1 X_{s1} + y_1 Y_{s1} + Z_{s1} \quad (14)$$

$$c = X_{s1}^2 + Y_{s1}^2 + Z_{s1}^2 - r_s^2 \quad (15)$$

où  $(x_1, y_1)$  dénote les coordonnées métriques de  $\mathbf{p}_1$  dans le repère de la caméra 1.

Des objets sphériques ont été considérés comme on le montrera dans la Section résultat. D'autres formes seront considérées à l'avenir.

#### 4.1. Matrice d'interaction

La matrice d'interaction  $\mathbf{L}_{I(\mathbf{p}_2)}$  est la matrice qui lie la variation de la valeur des niveaux de gris au mouvement de la caméra. Ici, la dérivation est donnée par [6] :

$$\mathbf{L}_{I(\mathbf{p}_2)} = \frac{\partial I(\mathbf{p}_2)}{\partial \mathbf{r}} = \nabla_{\mathbf{x}} \mathbf{I}_2^\top(\mathbf{p}_2) \frac{\partial \mathbf{p}_2}{\partial \mathbf{r}}, \quad (16)$$

où  $\nabla_{\mathbf{x}} \mathbf{I}_2^\top(\mathbf{y})$  est le gradient spatial de l'image  $\mathbf{I}_2$  à la position  $\mathbf{y}$  et  $\frac{\partial \mathbf{p}_2}{\partial \mathbf{r}} = \mathbf{L}_{\mathbf{p}_2}$  est la matrice d'interaction d'un point

de l'image exprimé en coordonnées pixelliques.  $\mathbf{L}_{\mathbf{p}_2}$  est donnée par :

$$\mathbf{L}_{\mathbf{p}_2} = \begin{pmatrix} f_x & 0 \\ 0 & f_y \end{pmatrix} \cdot \begin{pmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & (1+y^2) & -xy & -x \end{pmatrix} \quad (17)$$

$f_x$  et  $f_y$  sont les rapports focaux de la caméra et  $(x, y)$  dénote les coordonnées métriques du point  $\mathbf{p}_2$  dont la position en pixel dans l'image est donnée par  $(\frac{{}^2u}{Z}, \frac{{}^2v}{Z})$ .

L'information  $Z$  de profondeur est calculée à chaque itération. Par exemple pour un plan, on a :  $1/Z = \frac{{}^1d - {}^2\mathbf{t}_1^\top {}^1\mathbf{n}}{({}^2\mathbf{R}_1 {}^1\mathbf{n})^\top [x, y, 1]^\top}$  (voir (12) pour une structure sphérique).

#### 4.2. Le modèle de l'objet

L'estimation de déplacement a été présentée pour deux images  $\mathbf{I}_1$  et  $\mathbf{I}_2$ . Dans la pratique,  $\mathbf{I}_2$  est l'image courante pour laquelle la pose de la caméra doit être estimée et  $\mathbf{I}_1$  une image de référence du plan suivi. Il y a une image de référence pour chaque plan  $\pi_i$  avec la texture à suivre sur l'objet. Le modèle de l'objet se compose alors d'un modèle de DAO pour la partie basée contour du suivi et d'images de référence pour celle basée texture. Un calcul de pose est exécuté pour chaque image de référence en utilisant le suivi basé contour pour obtenir les paramètres du plan dans le repère de la caméra nécessaires pour (9) et le calcul de profondeur.

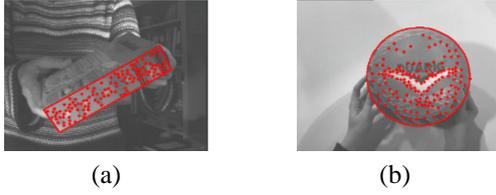
Si plusieurs plans sont suivis, le nombre d'échantillons de niveaux de gris par plan doit être mis à jour à chaque image puisque la visibilité de chaque plan évolue. S'il y a  $n_t$  échantillons de niveaux de gris considérés dans le processus de suivi, alors le nombre d'échantillons  $n_{t_i}$  de niveaux de gris pour le plan  $\pi_i$  est :

$$n_{t_i} = \frac{n_t}{\sum_i a_i} a_i, \quad (18)$$

où  $a_i$  est l'aire occupée par le plan  $\pi_i$  dans l'image,  $a_i$  étant égale à 0 si le plan  $\pi_i$  n'est pas visible. Pour chaque image de référence, les points de  $n_t$  sont sous-échantillonnés suivant un compromis entre le critère de Harris et une couverture maximale du motif afin d'améliorer la robustesse du suivi [22]. Dans la Figure 1, un exemple est donné pour chaque objet suivi dans la Section expérience. Selon la visibilité du plan, un ensemble de ces échantillons sera mis à jour et suivi suivant la règle indiquée en (18).

#### 4.3. Détection des données aberrantes

Puisque les niveaux de gris sont échantillonnés sur des points de Harris, *ie.* sur des zones de forts gradients lu-



**Figure 1. Modèle des motifs d'une face pour: (a) une boîte de riz, (b) un ballon**

mineux, un petit mouvement de la caméra peut entraîner un grand changement d'intensité lumineuse dans l'image. Pour éviter l'élimination systématique des points les plus intéressants du modèle, c'est le vector  $\Delta'$  normal suivant qui est utilisé pour calculer les M-estimateurs à la place de celui défini par (1) :

$$\Delta' = \left( \dots, \frac{I_1(\mathbf{p}_{1_i}) - I_2(2tr_1(\mathbf{p}_{1_i}))}{\|\nabla I_1(\mathbf{p}_{1_i})\|}, \dots \right) \quad (19)$$

En outre, l'illumination globale des les images de référence peut être différent quand le suivi est effectué. Pour améliorer le processus de M-estimation, la moyenne des niveaux de gris d'une image de référence est mise à jour quand le plan associé devient visible.

#### 4.4. Fusion des primitives basées contour et basées texture

Comme il a été dit, n'importe quel genre de primitives peut être considéré dans l'approche proposée en utilisant (4). Si des primitives basées contour et basées texture ainsi que leur matrice d'interaction associée sont empilées comme il est fait dans notre suivi hybride, une normalisation doit être effectuée pour tenir compte de l'information fournie par les différentes primitives. En effet, l'erreur liée à un point de texture (valeur de niveau de gris) et celle liée à un point de contour (distance de point-à-contour) sont d'un ordre de grandeur différent.

Par conséquent, l'ensemble des erreurs liées à une primitive basée contour (resp. une valeur de niveaux de gris) est normalisé de façon à ce que ces valeurs appartiennent à l'intervalle  $[-1; 1]$ .

Le suivi basé contour est nécessaire pour initialiser le suivi basé texture. Après, si seules les primitives basée texture sont exploitées dans la loi de commande pour estimer la pose de la caméra, le calcul de pose est relativement robuste si l'objet est donné à des motifs bien prononcés. Il est cependant sensible aux changements d'échelle et d'illumination. Il faut noter que dans le cas où seules des primitives basées texture sont employées, le cadre décrit est semblable à celui proposé pour le suivi 2D par [6] et

étendu pour le 3D par [12]. Une différence notable est que dans [12], le pseudo-inverse de la Jacobienne utilisée dans (3) est apprise lors d'une étape hors-ligne.

## 5. Expériences et résultats

Cette Section présente quelques résultats de suivi où notre suivi hybride est comparé au suivi basé contour et basé texture. Deux derniers utilisent dans le processus de suivi uniquement le type de primitives associé. La première expérience traite une structure planaire par morceaux et la deuxième un ballon.

Dans la première image des expériences de suivi, les points de contour et de texture utilisés dans le processus de minimisation sont montrés (pour les données correctes : les croix bleues pour les niveaux de gris et des croix rouges pour les endroits de contour. Ces derniers peuvent apparaître en noir si aucun contour n'est détecté. Les données erronées sont représentées par les croix vertes). La position d'objet dans chaque image est donnée par le contour courant en vert.

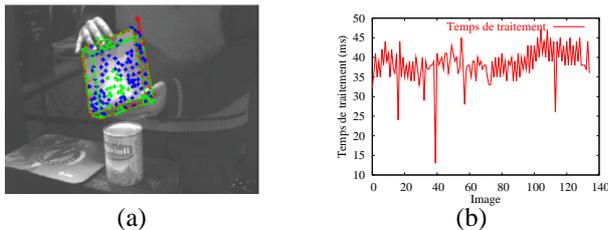
Pendant ces expériences, le suivi basé contour ou/et basée texture peut échouer tandis que l'hybride réussit. Le processus de M-estimation permet au suivi hybride d'être au moins aussi bon que l'un des deux suivis de base.

La caméra utilisée dans ces expériences est une caméra CCD monochrome. La cadence d'acquisition est de 25 Hz.

### 5.1. Séquence de la boîte de riz

Cette séquence vidéo pendant laquelle une boîte à riz est suivie est difficile puisque l'objet tourne sur lui-même. Par conséquent, les primitives suivies changent comme les faces apparaissent ou disparaissent. Si le suivi commence à diverger, il peut être difficile de rectifier l'erreur, d'autant plus que les positions des lumières entraînent des reflets assez importants. Les contours de l'objet sont en permanence cachés par les mains ou à peine visibles : le suivi basé contour finit par perdre l'objet (Figure 3(a)). Le facteur d'échelle de l'objet dans l'image est différente de celui dans les images de référence, ce qui mène le suivi basée texture à échouer très rapidement (Figure 3(b)). Cependant, alors que si les suivis de base ne suffisent pas pour un bon suivi, leur fusion dans le suivi hybride permet de suivre l'objet correctement (Figure 3(c)).

La Figure 2(a) montre un exemple de reflet qui peut poser problème au suivi, puisque que cela génère des données aberrantes. On voit bien que celles-ci, dessinées par des points verts, ont bien été détectées - et donc retirées du processus. Le suivi hybride tourne à une fréquence moyenne de 25 hertz. Le temps de suivi est donné dans Figure 2(b) : le suivi hybride proposé dans cet article est assez rapide pour un suivi en ligne.



**Figure 2. Séquence de la boîte de riz. (a): exemple de reflets. (b): évolution du temps de suivi.**

## 5.2. Séquence du ballon

La difficulté de cette expérience réside dans le fait que l'objet suivi est un ballon (reflets permanents,...). Le suivi basé contour (Figure 3(d)) réussit à suivre les contours de la balle mais ne fournit aucune information sur l'orientation de celle-ci (on peut voir le repère de l'objet qui reste fixe alors que la balle tourne). Le suivi basé texture finit par perdre la balle (Figure 3(c)) à cause des mauvaises conditions d'illumination. Le suivi hybride réussit non seulement à donner la position du ballon tout au long de la séquence mais donne aussi son orientation (le repère de l'objet bouge avec la balle) (Figure 3(d)).

## 6. Conclusion et perspectives

Un nouvel algorithme hybride a été construit à partir de deux suivis basés modèle classiques, exploitant l'extraction de contour et l'information portée par les motifs texture pour obtenir un calcul de pose plus robuste et plus précis de pose. L'intégration de l'estimation basée texture du mouvement de la caméra dans l'estimation basée contour de sa pose dans un même processus en utilisant le cadre de l'asservissement visuel virtuel permet un suivi temps réel nécessitant un modèle DAO et un modèle de la texture de l'objet. Les M-estimateurs sont ajoutés dans le processus de suivi pour améliorer la robustesse de l'algorithme aux occultations, aux ombres, aux reflets et au bruit.

Nous sommes maintenant intéressés à étendre ce suivi spatio-temporel à d'autres structures non-planaires pour agrandir l'éventail des objets pouvant être considérés. Puisque n'importe quelle amélioration du traitement d'un type de primitive dans le processus de suivi mène également à un meilleur suivi hybride, nous étudierons un modèle multi-échelle de la texture des faces pour améliorer la robustesse aux changements d'échelle.

## References

[1] P. Boutheymy. A maximum likelihood framework for determining moving edges. *IEEE PAMI*'89, 11(5):499–511, May 1989.  
 [2] A. Comport, E. Marchand, and F. Chaumette. A real-time tracker for markerless augmented reality. In *ACM/IEEE ISMAR'03*, pages 36–45, Tokyo, Japan, Oct. 2003.

[3] D. Dementhon and L. Davis. Model-based object pose in 25 lines of codes. *IJCV*'95, 15:123–141, 1995.  
 [4] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE PAMI*'02, 27(7):932–946, July 2002.  
 [5] D. Gennery. Visual tracking of known three-dimensional objects. *IJCV*'92, 7(3):243–270, 1992.  
 [6] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE PAMI*'98, 20(10):1025–1039, Oct. 1998.  
 [7] G. Hager and K. Toyama. The XVision system: A general-purpose substrate for portable real-time vision applications. *CVIU*'98, 69(1):23–37, Jan. 1998.  
 [8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2001.  
 [9] P.-J. Huber. *Robust Statistics*. Wiley, New York, 1981.  
 [10] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE ITRA*'96, 12(5):651–670, Oct. 1996.  
 [11] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *ECCV'96*, pages 343–356, Cambridge, UK, 1996.  
 [12] F. Jurie and M. Dhome. Read time 3D template matching. In *CVPR'01*, volume 1, pages 791–796, Hawaii, Dec. 2001.  
 [13] C. Kervrann and F. Heitz. A hierarchical Markov modeling approach for the segmentation and tracking of deformable shapes. *GMIP*, 60(3):173–195, May 1998.  
 [14] H. Kollnig and H.-H. Nagel. 3D pose estimation by fitting image gradients directly to polyhedral models. In *IEEE ICCV'95*, pages 569–574, Boston, MA, May 1995.  
 [15] V. Lepetit, L. Vacchetti, T. Thalmann, and P. Fua. Fully automated and stable registration for augmented reality applications. In *ACM/IEEE ISMAR'03, ISMAR'03*, pages 93–102, Tokyo, Japan, Oct. 2003.  
 [16] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE PAMI*'91, 13(5):441–450, May 1991.  
 [17] E. Marchand, P. Boutheymy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2D-3D model-based approach. In *IEEE ICCV'99*, volume 1, pages 262–268, Kerkira, Greece, Sept. 1999.  
 [18] E. Marchand and F. Chaumette. Virtual visual servoing: a framework for real-time augmented reality. In *EUROGRAPHICS'02*, volume 21(3) of *Computer Graphics Forum*, pages 289–298, Saarebrücken, Germany, Sept. 2002.  
 [19] M. Pressigout and E. Marchand. Model-free augmented reality by virtual visual servoing. In *IAPR Int. Conf. on Pattern Recognition, ICPR'04*, volume 2, pages 887–891, Cambridge, UK, Aug. 2004.  
 [20] M. Pressigout and E. Marchand. A model free hybrid algorithm for real time tracking. In *IEEE ICIP'05*, Genoa, Italia, September 2005.  
 [21] A. Shahrokni, T. Drummond, and P. Fua. Texture boundary detection for real-time tracking. In *ECCV'04*, volume 2, pages 566–577, Prague, Czech Republic, May 2004.  
 [22] J. Shi and C. Tomasi. Good features to track. In *IEEE CVPR'94*, pages 593–600, Seattle, Washington, June 1994.  
 [23] G. Simon and M.-O. Berger. Pose estimation for planar structures. *IEEE CGA'02*, 22(6):46–53, Nov. 2002.  
 [24] G. Taylor and L. Kleeman. Fusion of multimodal visual cues for model-based object tracking. In *ACRA'03*, Brisbane, Australia, December 2003.

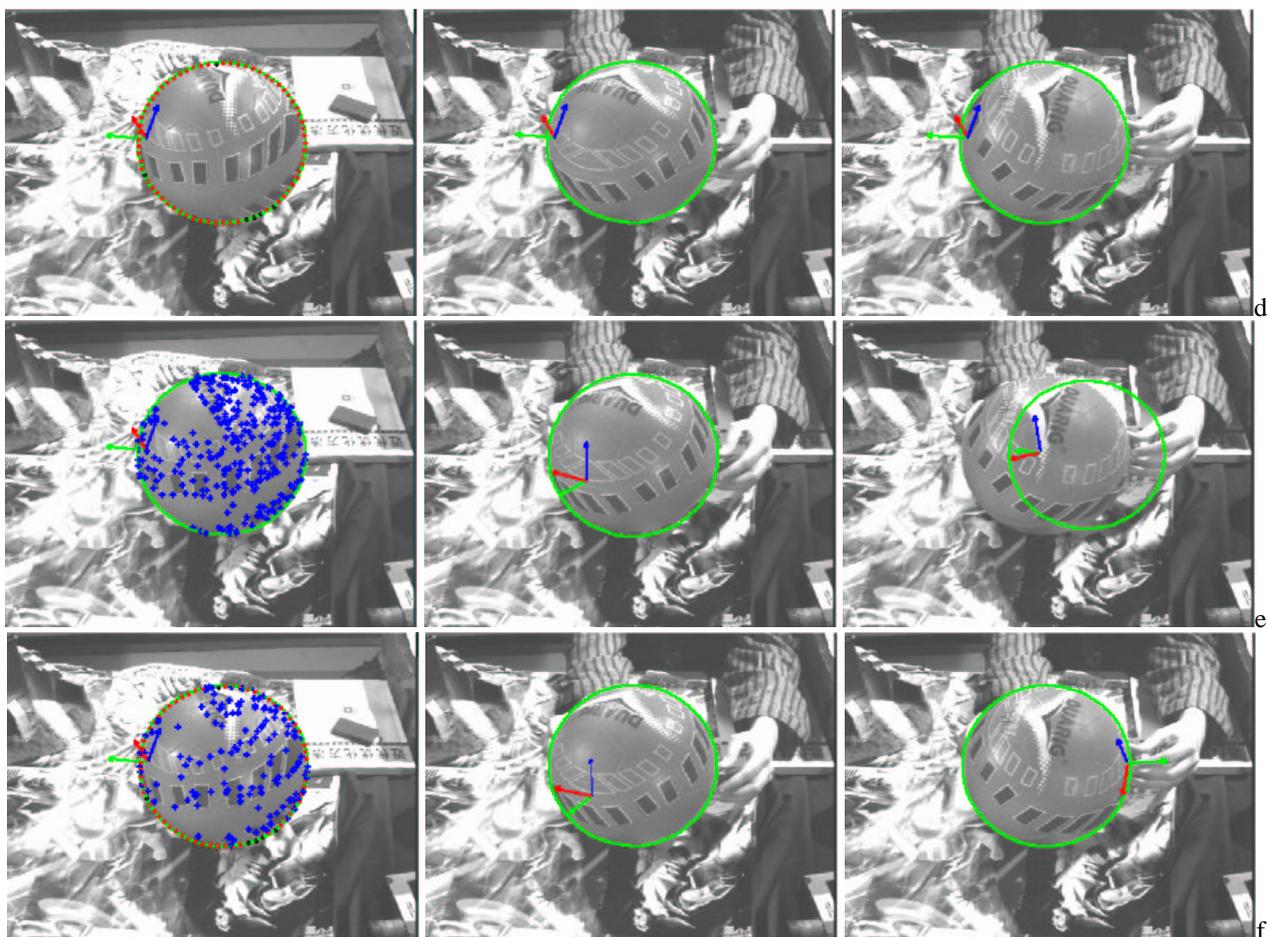


Figure 3. Images pour la première (boite de riz) puis pour le deuxième (ballon) expérience. Images pour (a), (d): le suivi basé contour, (b), (e): celui basé texture, (c), (f): le suivi hybride. Seul le suivi hybride réussit à suivre correctement les objets tout au long de la séquence, malgré les reflets et l'environnement. Le suivi basé contour suit bien l'enveloppe du ballon mais ne donne aucune indication sur son orientation.