Fusion de primitives visuelles pour le suivi 3D temps-réel

Muriel Pressigout, E. Marchand

Fusion de primitives visuelles pour le suivi 3D temps-réel
Fusion of visual features for real-time 3D tracking

M. Pressigout[1] É. Marchand[2]

[1] IRISA/INRIA
[2] Université de Rennes I

Campus de Beaulieu, 35042 Rennes Cedex
mpressig@irisa.fr,marchand@irisa.fr

Résumé

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 d'estimation et un modèle multi-échelle des motifs. L'approche a été validée sur des séquences vidéos ainsi que sur des applications en ligne de Réalité Augmentée. Ces résultats montrent que la méthode proposée permet une localisation à la cadence vidéo tout en étant robuste aux larges mouvements et à des environnements réalistes.

Mots Clef

Suivi basé modèle, fusion texture et contour, approche multi-échelle, temps-réel, estimation robuste

Abstract

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 using a multi-scale model of the texture and 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 considering various objects on several video sequences as well as during Augmented Reality experiments. Results show the method to be robust to large motions and textured environments.

Keywords

Model-based tracker, texture and contour fusion, multi-scale approach, real-time, robust estimation

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. Cette estimation nommée calcul de pose se base sur les informations visuelles extraites des images acquises. Les applications sont diverses et variées et vont de la robotique à 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 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 robuste.

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'interêt [7]... La seconde utilise en plus explicitement un modèle des objets suivis pouvant être un modèle 3D CAO [4, 5, 14, 16] ou un modèle 2D de l'objet [13]. Cette dernière 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.

Suivis basés sur les contours. 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 c.à.d. 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 consi-

dérées 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 images 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, si l'environnement ou l'objet comporte des motifs ambigus, le suivi peut devenir irrégulier et donner une mauvaise estimation de la pose.

Suivis basés sur la texture. L'information portée par les motifs de l'objet est utilisée dans beaucoup de travaux pour le suivi d'objets dans des séquences d'image. Contrairement aux algorithmes basés sur les contours de l'objet, elle est bien adaptée aux objets texturés et ne souffre généralement pas d'irrégularités dans l'estimation des paramètres. Elle peut être utilisée pour le suivi de région d'intérêt [6, 12] ou le suivi de points d'intérêts, tel que l'algorithme KLT [22]. Les points d'intérêt peuvent également être utilisés pour le suivi 3D dans la phase de recalage 2D-3D [19, 23]. Dans [26], les points d'intérêts sont utilisés dans deux processus fusionnés. Le premier fournit une pose pour chaque image en effectuant un recalage 2D-3D par rapport à des images clés et le second un recalage 2D-2D avec les images précédentes pour imposer une contrainte temporelle qui lie les images d'une même scène. 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 et ne peut être appliquée si l'objet est peu texturé. Dans ce cas, une approche multiéchellle peut être effectuée.

Suivis hybrides. 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 réaliste, faire la distinction entre chaque cas n'est pas évident. 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. Dans des travaux antérieurs [20], nous avons proposé 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 contour. Fusionner les deux approches pour effectuer un suivi 3D a été étudié dans les travaux récents de [25, 15]. [25] 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. Dans [15], l'approche proposée dans [26] est étendue en fusionnant les processus basés sur les points d'intérêts avec un recalage 2D-3D par reprojection du modèle 3D en considérant des hypothèse multiples pour le suivi des contours.
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. De plus, le modèle des motifs permet de prendre en compte les changements d'échelle sans perturber le suivi.
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 intensivement utilisé, le suivi bayésien est une technique très différente de celle présentée dans cet article et quoique les objectifs soient semblables, l'aspect théorique est très différent et peut à peine être comparé.

Principe de notre approche Dans cet article, la pose et le calcul de déplacement de la caméra formant la base du suivi 3D 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, 18]. 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 tâches de positionnement ou de suivi) est alors spéficiée 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 robustesse). Pour améliorer la robustesse, un M-estimateur est intégré dans la loi de commande. L'algorithme de calcul de pose ou de déplacement résultant peut ainsi considéré efficacement les primitives bas niveau dont le suivi est erronné sans dégrader le comportement de l'algorithme.
Dans la suite de cet article, 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. Finalement, afin de valider cette approche l'algorithme est testé sur plusieurs séquences vidéo réalistes. Ces résultats expérimentaux sont rapportés dans la section 5.

2 Suivi basé modèle: un problème de recalage 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'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. L'approche consiste à estimer la vraie pose ou le vrai déplacement en minimisant l'erreur Δ entre les données observées $\mathbf{s}^*$ et la valeur courante $\mathbf{s}$ des mêmes primitives calculées par une projection selon la pose ou le déplacement courant :

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

où $\rho(u)$ est une fonction robuste [9] et $\mathbf{q}$ sont les paramètres courants de la pose ou du déplacement selon le problème à résoudre. Cette formulation de l'erreur est utilisée à la fois pour le calcul de pose de la caméra et le calcul de déplacement, toutefois les primitives $\mathbf{s}$ 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 Δ. À la convergence, la caméra virtuelle a réalisé le positionnement ou le déplacement qui minimise l'erreur Δ.
Cet objectif est incorporé à une loi de commande robuste. Δ est alors minimiser en utilisant un processus de moindre carré pondéré itéré (IRLS: Iterated Reweighted Least Square). L'erreur régulée à 0 est alors définie par :

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

où $\mathbf{D}$ est une matrice de poids diagonale donnée par $\mathbf{D} = diag(w_1, \ldots, w_k)$. Les poids w_i reflètent la confiance en chaque primitive et leur calcul est basé sur les M-estimateurs. Le lecteur pourra se référer à [2] pour une utilisation des m-estimateurs dans ce contexte ou à [9] pour un cadre général. Une loi simple de commande peut alors être conçue pour assurer une diminution exponentielle de $\mathbf{e}$ autour de la position désirée $\mathbf{s}^*$. La loi de commande est donnée par [2]:

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

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

Une grande variété de primitives peut être considérée dans la loi de commande proposée dès qu'il est possible de calculer sa matrice d'interaction correspondante $\mathbf{L}_\mathbf{s}$. La combinaison de différentes primitives est réalisée en ajoutant des primitives au vecteur $\mathbf{s}$ 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 :

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

Selon la nature des primitives, les techniques d'AVV [18, 24] 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 dans un contexte multi-image. La section suivante est consacrée au choix des primitives visuelles et à leur rôle dans la méthode proposée.

3 Choix des primitives visuelles

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 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 Δ entre les données observées $\mathbf{s}^*$ (la position d'un ensemble de primitives dans l'image dans le cas du calcul de pose) et la position $\mathbf{s}$ de ces mêmes primitives calculée par une projection selon les paramètres courants de la pose :

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

où $pr_\xi(\mathbf{r}, {}^o\mathbf{P})$ est le modèle de projection dépendant des paramètres intrinsèques ξ 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 chaque axe. On suppose ici que les paramètres intrinsèques ξ 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 Δ ($\mathbf{r}$ 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 différentes 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 CAO reprojeté 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 décrits par morceaux linéaires, toutes les distances sont traitées en fonction du segment associé.

Le suivi basé contour correspond à une méthode classique de calcul de pose. Il est rapide, efficace et robuste aux changements d'illumination. Cependant, c'est principalement un processus mono-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[1]. Dans les deux cas, la vitesse de la caméra est calculée pour mettre à jour la pose ou le déplacement de la caméra, ce qui revient au même puisque :

$$ {}^2\mathbf{M}_o = {}^2\mathbf{M}_1 . {}^1\mathbf{M}_o \tag{6}$$

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 notée ${}^2tr_1$. ${}^2tr_1$ dépend du déplacement de la caméra et des contraintes géométriques entre plusieurs vues d'une même scène. L'équation (1) est alors :

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

où N est le nombre de pixels pris en compte. Plus de détails concernant la transformation 2D et la matrice,... seront vus par la suite. À la convergence, la caméra virtuelle a réalisé le déplacement $\widehat{{}^2\mathbf{M}_1}$ qui minimise cette erreur ($\widehat{{}^2\mathbf{M}_1}$ 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 Estimation du déplacement de la caméra: comment faire?

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 ensuite.

4.1 Transformation 2D

Structure planaire. Supposons dans un premier temps que le motif à suivre soit 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 :

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

où $\mathbf{K}$ est la matrice des paramètres intrinsèques de la caméra et ${}^2\mathbf{H}_1$ est une homographie (définie à un facteur d'échelle près) qui définit la transformation en coordonnées métriques entre les images acquises par la caméra aux poses 1 et 2. L'homographie ${}^2\mathbf{H}_1$ est donnée par :

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

où ${}^1\mathbf{n}$ et ${}^1d$ sont respectivement la normale du plan et la distance signée 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] :

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

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

où ${}^2\mathbf{H}_1$ est l'homographie induite par un plan de référence π comme vu dans le paragraphe précédent, le scalaire β_1 la parallaxe relative à l'homographie ${}^2\mathbf{H}_1$ et $\mathbf{c}_2 = \mathbf{K}^2\mathbf{t}_1$ l'épipole projeté sur l'image 2 en coordonnées pixelliques. β_1 peut être interprétée comme une profondeur relative au plan π :

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

avec Z_1 la coordonnée en profondeur du point 3D associée à $\mathbf{p}_1$ exprimée dans le repère 1 de la caméra.
Puisque que β_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 $\mathbf{p}_1$. Ainsi, par exemple, dans le cas d'une sphère dont le rayon est r_s et le centre $\mathbf{c}_{s1} = (X_{s1}, Y_{s1}, Z_{s1})$, on a :

$$Z_1 = \frac{b - \sqrt{b^2 - ac}}{a} \tag{12}$$

avec :

$$a = x_1^2 + y_1^2 + 1 \tag{13}$$
$$b = x_1 X_{s1} + y_1 Y_{s1} + Z_{s1} \tag{14}$$
$$c = X_{s1}^2 + Y_{s1}^2 + Z_{s1}^2 - r_s^2 \tag{15}$$

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.

4.2 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] :

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

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 $\frac{\partial \mathbf{p}_2}{\partial} = \mathbf{L}_{\mathbf{p}_2}$ est la matrice d'interaction d'un point de l'image exprimé en coordonnées pixelliques. $\mathbf{L}_{\mathbf{p}_2}$ est donnée par :

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

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 $(\frac{^2u}{^2w}, \frac{^2v}{^2w})$.
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).

4.3 Le modèle multi-échelle 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 π_i texturé à suivre. Le modèle de l'objet se compose alors d'un modèle de CAO 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 π_i est :

$$n_{t_i} = \frac{n_t}{\sum_i a_i} a_i, \tag{18}$$

où a_i est l'aire occupée par le plan π_i dans l'image, a_i étant égale à 0 si le plan π_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).

(a) (b) (c)

FIG. 1 – *Modèle des motifs d'une face pour: (a) une boîte de riz, (b) une boîte de DVDs et (c) un ballon*

Ce modèle des niveaux de gris décrit bien la texture de l'objet pour un intervalle de distance caméra-objet donné. La génération d'une pyramide d'images de référence va permettre de décrire les niveaux de gris qui représentent mieux l'objet à une distance plus importante. À partir de l'image de référence $\mathbf{I}_{1i}$ associée à un plan π_i, K images $\mathbf{I}_{1i}^k$vont être construites par un filtrage gaussien et sous-échantillonnage. L'image $\mathbf{I}_{1i}^k$ est obtenue à partir de $\mathbf{I}_{1i}^{k-1}$ par :

$$\mathbf{I}_{1i}^k = f(\mathbf{G} \otimes \mathbf{I}_{1i}^{k-1}) \tag{19}$$

où f est la fonction de sous-échantillonnage (un pixel sur deux), $\mathbf{G}$ un filtre gaussien et $\otimes$ l'opérateur de convolution. Le point de départ est donné par l'image de référence $\mathbf{I}_{1i} = \mathbf{I}_{1i}^1$. Seule une des images $\mathbf{I}_{1i}^k$ sera suivie si le plan π_i

est visible. Le choix se fera en fonction de la distance objet-caméra estimée pour l'image précédente de la séquence. La Figure 2 montre un exemple de pyramide pour la face de la boîte de DVD sur laquelle va être effectuée une détection de points de Harris à chaque niveau.


a b c

FIG. 2 – *Modèle pyramidale d'une face. (a) niveau 1, (b) niveau 2, (c) niveau 3*

4.4 Détection des données aberrantes

Puisque les niveaux de gris sont échantillonnés sur des points de Harris, c.à.d. sur des zones de forts gradients lumineux, 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 l'erreur Δ' suivante qui est minimisée à la place de celle définie dans (7) :

$$\Delta' = \sum_{i=1}^{N} \rho\left(\frac{I_1(\mathbf{p}_{1_i}) - I_2({}^2tr_1(\mathbf{p}_{1_i}))}{\| \nabla I_1(\mathbf{p}_{1_i}) \|}\right) \qquad (20)$$

En outre, l'illumination globale des images de référence peut être différent quand le suivi est effectué. Pour améliorer le processus de M-estimation, il faut prendre en compte les données observées dans les images précédentes et mettre à jour les niveaux de gris d'une image de référence quand le plan associé devient visible puis de façon régulière.

4.5 Fusion des primitives basées contour et basées texture

Comme il a été dit, une grande variété de primitives peut être considérée 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 $[-1; 1]$.

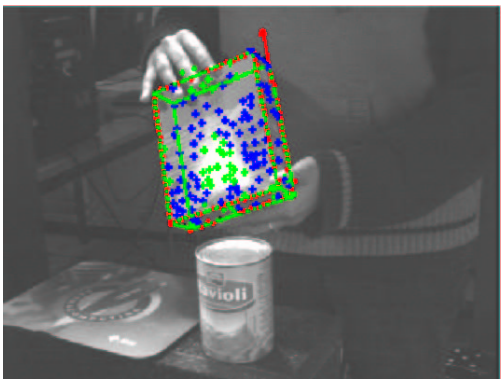
Le suivi basé contour est nécessaire pour initialiser le suivi basé texture. Ensuite, si seules les primitives basées texture sont exploitées dans la loi de commande pour estimer la pose de la caméra, le calcul de pose est relativement robuste si l'objet est donné à des motifs bien prononcés. Il est cependant sensible 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

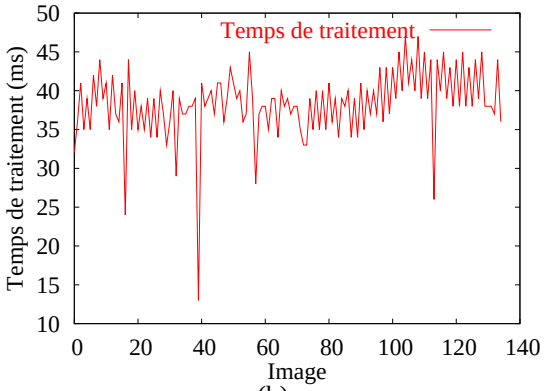
Cette section présente quelques résultats de suivi où notre suivi hybride est comparé aux suivis basé contour et basé texture. Les deux derniers utilisent dans le processus de suivi uniquement les primitives associées. Les deux premières expériences considèrent une structure planaire par morceaux et la troisième un ballon. La caméra utilisée dans ces expériences est une caméra CCD monochrome.
Dans la première image de chacune des expériences de suivi, les points de contour et de texture utilisés dans le processus de minimisation sont présentés. 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, c.à.d. les outliers détectés par les M-estimateurs, sont représentées par les croix vertes. La position de l'objet dans chaque image est donnée par le contour courant en vert.
Pendant ces expériences, le suivi basé contour et/ou basée texture peut échouer tandis que l'hybride réussit. Le processus de minimisation robuste permet au suivi hybride d'être au moins aussi bon que l'un des deux suivis de base.

5.1 Séquence de la boite 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 souvent du fait de l'apparition et de la disparition des faces. Si le suivi commence à diverger, il peut être difficile de rectifier l'erreur, d'autant plus que les positions des lumières entraînent des spécularités assez importantes. 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 5(a)). Le suivi basée texture échoue à se recaler à cause du retournement de l'objet (Figure 5(b)). Cependant, alors que si les suivis de base ne suffisent pas pour effectuer un bon suivi, leur fusion dans le suivi hybride décrit ans cet article permet de suivre l'objet précisément (Figure 5(c)).
La Figure 3(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 ont bien été détectées (croix vertes) et donc retirées du processus. Le suivi hybride tourne à une fréquence moyenne de 25 hertz. Le temps de traitement pour chaque image est donné dans la Figure 3(b) : le suivi hybride proposé dans cet article est assez rapide pour un suivi en ligne.



(a)

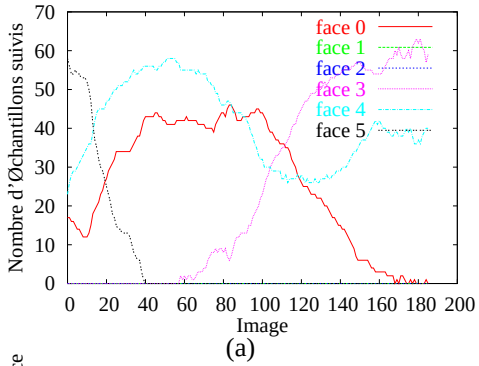


(b)

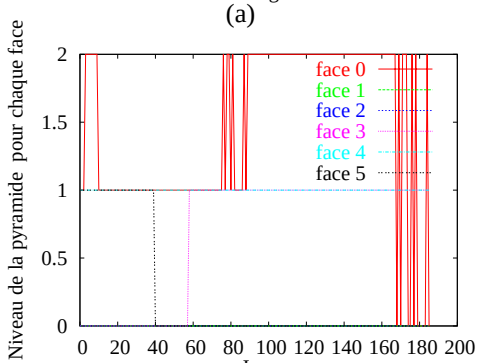
FIG. 3 – *Séquence de la boite de riz. (a): exemple de spécularités. Au centre de celle-ci, les primitives sont rejetées automatiquement par les M-estimateurs. (b): évolution du temps de suivi.*

5.2 Séquence de la boite de DVD

Cette expérience a pour but de montrer le mécanisme et l'intérêt du modèle multi-échelle des motifs. L'objet suivi est une boîte de DVD. Dans un premier temps, le modèle de texture considéré est constitué d'une image de référence par face de l'objet (Figure 6(a)) puis ensuite, l'approche pyramidale est utilisée (Figure 6(b)). Elle permet d'effectuer un suivi correct, ce qui n'est pas le cas pour la première option. Les Figures 4(a) et 4(b) représentent respectivement l'évolution du nombre d'échantillons suivis par face selon (18) et le choix du motif dans la pyramide des images de références de chaque face tout au long de la séquence. En ce qui concerne ce dernier graphe, le niveau 0 signifie que la face n'est pas suivie, le niveau 1 correspond au motif de référence le plus grand de la pyramide (celui qui a servi à construire cette dernière), le niveau 2 correspondant au motif sous-échantillonné une fois.
L'utilisation d'un modèle multi-échelle permet d'améliorer la qualité du suivi et ceci sans changement dans le temps de traitement d'image. La vitesse de l'algorithme n'est pas modifiée puisque le nombre de primitives suivies est identique, seule leur valeur de référence diffère.



(a)



(b)

FIG. 4 – *Séquence de la boîte de DVD. (a) nombre d'échantillon de niveaux de gris pris en compte par face, (b)*

5.3 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 5(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). En effet, seule la silhouette du ballon est prise en compte, ce qui permet d'estimer uniquement les degrés de liberté en translation mais pas en rotation. Le suivi basé texture finit par perdre la balle (Figure 5(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 (visuellement on peut constater que le repère associé de l'objet bouge de façon cohérente avec la balle) (Figure 5(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. L'intégration de l'estimation du mouvement de la caméra basée sur la texture avec l'estimation de sa pose basée sur les contours au sein d'un même processus en utilisant le cadre de l'asservissement visuel virtuel permet un suivi temps réel nécessitant un modèle CAO 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 par 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 de l'illumination de la texture des faces pour améliorer la robustesse aux changements d'illumination.

Références

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

[2] A.I. Comport, E. Marchand, et F. Chaumette. A real-time tracker for markerless augmented reality. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03*, pages 36–45, Tokyo, Japon, Octobre 2003.

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

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

[5] D.B. Gennery. Visual tracking of known three-dimensional objects. *Int. J. of Computer Vision*, 7(3):243–270, 1992.

[6] G. Hager et P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, Octobre 1998.

[7] G. Hager et K. Toyama. The XVision system: A general-purpose substrate for portable real-time vision applications. *Computer Vision and Image Understanding*, 69(1):23–37, Janvier 1998.

[8] R. Hartley et A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2001.

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

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

[11] M. Isard et A. Blake. Contour tracking by stochastic propagation of conditional density. In *European Conf. on Computer Vision, ECCV'96, LNCS no. 1064, Springer-Verlag*, pages 343–356, Cambridge, RU, 1996.

[12] F. Jurie et M. Dhome. Read time 3D template matching. In *Int. Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 791–796, Hawai, Décembre 2001.

[13] C. Kervrann et F. Heitz. A hierarchical Markov modeling approach for the segmentation and tracking of deformable shapes. *Graphical Models and Image Processing*, 60(3):173–195, Mai 1998.

[14] H. Kollnig et H.-H. Nagel. 3D pose estimation by fitting image gradients directly to polyhedral models. In *IEEE Int. Conf. on Computer Vision*, pages 569–574, Boston, MA, Mai 1995.

[15] V. Lepetit, L. Vacchetti, T. Thalmann, et P. Fua. Fully automated and stable registration for augmented reality applications. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03*, pages 93–102, Tokyo, Japon, Octobre 2003.

[16] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, Mai 1991.

[17] 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, Grèce, Septembre 1999.

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

[19] M. Pressigout et 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, RU, Août 2004.

[20] M. Pressigout et E. Marchand. A model free hybrid algorithm for real time tracking. In *IEEE Int. Conf. on Image Processing, ICIP'05*, Gênes, Italie, Septembre 2005.

[21] A. Shahrokni, To Drummond, et P. Fua. Texture boundary detection for real-time tracking. In *European Conf. on Computer Vision, ECCV'04, LNCS 3022*, volume 2, pages 566–577, Prague, République tchèque, Mai 2004.

[22] J. Shi et C. Tomasi. Good features to track. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, pages 593–600, Seattle, Washington, Juin 1994.

[23] G. Simon et M.-O. Berger. Pose estimation for planar structures. *IEEE Computer Graphics and Applications*, 22(6):46–53, Novembre 2002.

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

[25] G. Taylor et L. Kleeman. Fusion of multimodal visual cues for model-based object tracking. In *Australasian Conference on Robotics and Automation (ACRA2003)*, Brisbane,Australie, Décembre 2003.

[26] L. Vacchetti, V. Lepetit, et P. Fua. Stable 3–d tracking in real-time using integrated context information. In *IEEE Int. Conf. on Conf. on Computer Vision and Pattern Recognition, CVPR'03*, volume 2, pages 241–248, Madison, WI, Juin 2003.

FIG. 5 – *Séquence de la boîte de riz. Images pour les suivis (a): basé contour, (b): basé texture, (c): hybride. Seul le suivi hybride réussit à suivre correctement les objets tout au long de la séquence, malgré les reflets et l'environnment.*

FIG. 6 – *Séquence de la boîte de DVD. (a) avec une seule image de référence par face, (b) avec une pyramide d'images par face. Le modèle multi-échelle de la texture améliore les performances du suivi: il permet de prendre en compte la distance caméra-objet courante sans dégrader ralentir le traitement d'image.*

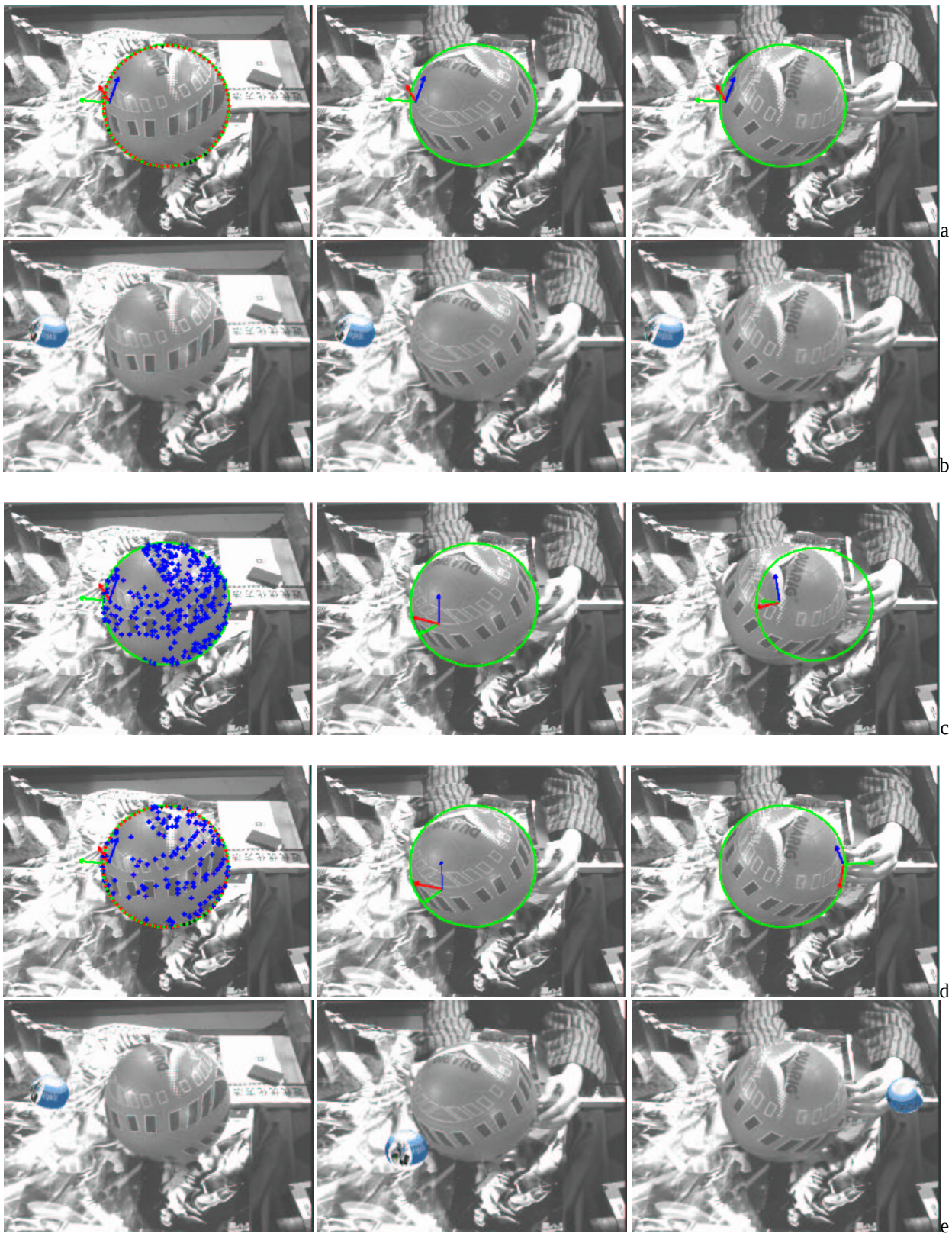


FIG. 7 – *Séquence du ballon . Images pour (a): le suivi basé contour, (b): la séquence augmentée en utilisant celui-ci. Comme on peut le voir, seule la position du ballon est estimée, l'orientation estimée du ballon reste identique bien qu'en réalité il tourne. , (c):le suivi basé texture , (d): le suivi hybride. Seul ce dernier résssuit à suivre complétement l'objet malgré les reflets, (e): la séquence augmentée en utilisant le suivi hybride: l'objet tourne autour du ballon lorsque le ballon tourne sur lui-même.*