

Utilisation d'un capteur inertielle comme aide au suivi basé vision

Use of Inertial Sensor to Support Video Tracking

Michael Aron

Gilles Simon

Marie-Odile Berger

LORIA - UHP Nancy 1 - INRIA Lorraine, France

{aron,gsimon,berger}@loria.fr

Résumé

L'un des principaux obstacles à l'obtention de systèmes de Réalité Augmentée effectifs est le manque de robustesse et/ou de précision des processus visant à acquérir la position de l'utilisateur au cours du temps. Cet article présente un système hybride, combinant un capteur inertielle à un système de suivi basé vision. Cette approche collaborative est motivée par le fait que les systèmes basés vision peuvent échouer lors du suivi, alors que les capteurs inertiels sont robustes, bien que peu précis. Notre approche présente plusieurs avantages : (i) elle prend en compte les erreurs de mesures du capteur, notamment lors de la phase d'appariement des primitives utilisées pour le suivi de la caméra. Ces erreurs sont analysées puis intégrées au suivi des points d'intérêt ; (ii) le capteur inertielle et la caméra sont synchronisés en ligne, et le capteur est utilisé seulement lorsque cela s'avère nécessaire. Un système effectif utilisé dans le cadre d'une application de commerce électronique est présenté à la fin de cet article.

Mots Clef

Réalité Augmentée, suivi basé vision, capteur inertielle, approche hybride

Abstract

One of the biggest obstacle to building effective augmented reality (AR) systems is the lack of accurate sensors that report the location of the user in an environment during arbitrary long periods of movements. In this paper, we present an effective hybrid approach that integrates inertial and vision based technologies. This work is motivated by the need to explicitly take into account the relatively poor accuracy of inertial sensors and thus to define an efficient strategy for the collaborative process between the vision based system and the sensor. The contributions of this papers are threefold : (i) our collaborative strategy fully integrates the sensitivity error of the sensor : the sensitivity is practically studied and is propagated into the collaborative process, especially in the matching stage (ii) we propose an original online synchronization process between the vision based system and the sensor. This process allows us to use

the sensor only when needed. (iii) an effective AR system using this hybrid tracking is demonstrated through an e-commerce application in unprepared environments.

Keywords

Augmented Reality, camera tracking, sensor fusion.

1 Introduction

La Réalité Augmentée (RA) a pour but d'améliorer notre perception du monde réel par l'ajout d'éléments virtuels. De nombreuses applications existent, par exemple dans la construction et la maintenance de machines complexes, la visualisation médicale ou encore le travail collaboratif [1]. Bien qu'ayant connu une phase de développement importante à la fin des années 1990, de nombreux problèmes subsistent toujours en RA. L'un des principaux enjeux actuels est celui du suivi, permettant de retrouver la position de la caméra. Connaissant cette position, les éléments virtuels peuvent être ajoutés pour un rendu précis et réaliste.

Le suivi est une tâche difficile dans le sens où il doit être effectué séquentiellement, le plus précisément possible, et être temporellement stable. Les systèmes permettant un suivi robuste sont donc un thème de recherche très actif qui permettront probablement l'apparition d'une multitude de nouvelles applications.

Quelle que soit la technique adoptée pour le suivi, la plupart des applications de RA exigent un environnement contrôlé ou restreignent les mouvements de l'utilisateur :

- le suivi à l'aide d'un capteur inertielle est souvent adopté car il est rapide et ne nécessite pas d'éléments extérieurs, mais il connaît souvent une large accumulation d'erreurs et une faible précision (entre 0.25 et 1 degré) ;
- les capteurs magnétiques et à ultra-sons ont aussi souvent été utilisés, mais ils possèdent l'inconvénient de confiner l'utilisateur dans un environnement petit et instrumenté ;
- les méthodes basées vision prennent en compte directement les propriétés de l'image : dans ce sens, elles permettent un recalage précis de la scène filmée et des éléments ajoutés dans cette scène. Ces méthodes empêchent cependant le suivi dans le cadre d'un mouve-

ment brusque de l'utilisateur, à cause du flou et de la disparité importante générées dans l'image par ce type de mouvement.

Utiliser seulement l'une des approches précédemment citées contraint l'utilisateur à restreindre ses mouvements, sans le laisser libre de ses déplacements. L'idée de faire collaborer deux de ces techniques n'est pas nouvelle et est apparue au début des années 1990 principalement en robotique. [2] ont abordé cette fusion pour des systèmes robotiques intelligents. Puis cette idée s'est peu à peu étendue à la Réalité Augmentée avec [3] et [4] en 1996, suivi de [5] et [6]. Dans la plupart des applications, la stratégie de collaboration entre le capteur et la caméra est la même pour toutes les images acquises : les données inertielles ou gyroscopiques sont utilisées pour prédire la position des points d'intérêt et rendre plus facile la détection et le suivi de ces points [6], [7]. On utilise aussi souvent un filtre de Kalman pour réduire l'erreur et fusionner les données. Malheureusement, ce filtre nécessite un mouvement régulier (à accélération constante par exemple, [2]) et ne permet donc pas de prendre en compte des mouvements brusques et inattendus. Nous avons décidé d'adopter une approche différente et nous utilisons le capteur seulement lorsque cela s'avère nécessaire, i.e. lorsque le système basé vision ne parvient plus à faire le suivi. Cette approche possède des similitudes avec celle adoptée en [8] dans le contexte de robots mobiles guidés par odométrie et capteurs gyroscopiques : le guidage est toujours basé sur l'odométrie ; seulement lorsque de mauvais résultats apparaissent, les capteurs gyroscopiques prennent le relais.

Notre méthode adopte aussi cette stratégie, et notre algorithme utilise les données du capteur inertiel seulement lorsque le système basé vision engendre de mauvais résultats. Comparée aux travaux existants, notre approche présente donc les caractéristiques suivantes :

- nous ne cherchons pas à combiner les données caméra/capteur en permanence. Les données capteur ne sont utilisées que pour assister le système basé vision en cas de besoin ;
- nous proposons un moyen de détecter automatiquement les défaillances du système basé vision, nécessitant de prendre en compte les données capteur,
- nous ne cherchons pas à corriger les erreurs d'accumulation du capteur au cours du temps car nous n'utilisons que les rotations relatives entre deux images,
- une synchronisation est effectuée entre la caméra et le capteur inertiel. C'est une étape particulièrement délicate mais importante puisqu'une mauvaise synchronisation engendrerait un décalage entre les données capteur et caméra, et ne permettrait donc pas de mettre en correspondance la rotation associée à l'image. Ce problème, relativement peu traité en RA, est particulièrement difficile à résoudre comme précisé en [6]. L'une des principales contributions de ce papier est de montrer que le délai d'acquisition des données entre le capteur et la caméra n'est pas constant dans le temps, et de proposer une

solution remédiant à ce problème.

L'efficacité de notre système hybride est démontrée à partir du système de suivi de plans que nous avons proposé dans [9], utilisé conjointement avec un capteur inertiel MT9 développé par la société XSens. Cependant, notre méthode est adaptée à n'importe quel système de suivi basé vision, pourvu qu'il respecte les contraintes suivantes : 1. la position de la caméra est obtenue à partir d'appariement de primitives suivies entre images consécutives de la séquence ; 2. un critère numérique permettant de juger si la position est calculée avec suffisamment de précision est disponible. Un exemple typique de tel critère est le nombre de primitives ayant été correctement suivies dans l'image courante (primitives inliers). Cette information est généralement obtenue en utilisant des M-estimateurs ou l'algorithme de RANSAC. De nombreux systèmes utilisant ces procédés peuvent être trouvés dans la littérature [10], [11].

Cet article est organisé de la façon suivante : la partie 2 s'intéresse à la précision du capteur. Dans la partie 3, nous étudions la méthode hybride pour terminer dans la partie 4 sur notre algorithme de synchronisation en temps réel de la caméra et du capteur inertiel. Un système complet a été développé dans le cadre du projet européen ARIS, pour lequel nous présentons nos résultats dans les parties 5 et 6.

2 Précision du capteur

Les récents progrès dans la miniaturisation permettent aujourd'hui à des sociétés comme XSens ou InterSense de produire des capteurs inertiels intégrant plusieurs gyroscopes, accéléromètres, magnétomètres et thermomètres dans de petits boîtiers ($39 \times 54 \times 28$ mm dans notre cas), pour des prix raisonnables. Ces capteurs mesurent les vitesses de rotation autour des 3 axes grâce aux gyroscopes. Des erreurs sont obtenues lors de l'intégration de ces vitesses de rotation, qui sont compensées par les données issues des accéléromètres et magnétomètres. Puisque les données produites ne dépendent que de phénomènes physiques naturels et n'exigent pas un contrôle de l'environnement, les capteurs inertiels sont parfaitement adaptés à des applications de RA en extérieur. Cependant, les données techniques fournies par les constructeurs indiquent souvent une précision de l'ordre de 1° RMS pour les produits les plus récents, ce qui est malheureusement encore insuffisant pour un suivi efficace.

Dans cette partie, nous présentons des tests de précision effectués sur le capteur inertiel XSens MT9-B, dans le but d'évaluer la précision de ce type d'appareil, et de vérifier si l'erreur obtenue sur les mesures des angles d'Euler peut être approximée par une loi gaussienne. Lors de nos expérimentations, le capteur est fixé sur une unité mobile (PTU) possédant 2 degrés de liberté et une précision de 0.0013° . Nous avons effectué trois séries de tests, en changeant à chaque fois l'orientation du capteur sur la PTU.

Pour chaque série, nous avons réalisé sur chaque axe de la PTU de nombreuses rotations (1000 rotations sur chacun des axes) de diverses amplitudes (2, 5, 10, 15, et 20 degrés).

La figure 1 présente les moyennes et écart-types obtenus. Les moyennes ne correspondent pas exactement aux commandes envoyées à la PTU, car l’alignement des axes du capteur avec ceux de la PTU n’est pas parfait. Cependant, on peut déduire de ces expérimentations deux principales informations : 1. la précision des mesures d’angles données par le capteur ne dépend pas de l’amplitude de la rotation et 2. la précision obtenue sur les angles pour un axe donné dépend de l’orientation de cet axe sur la PTU. Par exemple, les mesures sur l’axe x sont plus précises dans la configuration 1 que dans la configuration 2. Les rotations autour de l’axe vertical (dans le référentiel terrestre) du capteur sont toujours moins précises que des rotations autour des axes horizontaux. Ce phénomène est dû au fait que les magnétomètres qui compensent les erreurs des gyroscopes sur l’axe vertical sont beaucoup moins précis que les accéléromètres qui compensent les erreurs des gyroscopes sur les axes horizontaux. Les magnétomètres sont en effet très sensibles aux perturbations électromagnétiques de l’environnement, ce qui n’est pas le cas des accéléromètres.

En pratique, le capteur inertiel est fixé horizontalement sur la caméra, de manière à ce que l’axe z coïncide avec la verticale. Dans cette configuration, on s’attend à obtenir un bruit gaussien sur chacun des axes, de moyenne nulle et d’écart-types donnés par nos tests ($\sigma_z = 0.499^\circ$ et $\sigma_x = \sigma_y = 0.155^\circ$). Durant la phase de suivi, les écarts-types sont interpolés suivant les angles entre les axes considérés et l’axe z du capteur en sa position initiale.

3 Collaboration Caméra-capteur

3.1 Calibration hand-eye

Afin d’utiliser le capteur inertiel avec la caméra, nous avons besoin de connaître l’alignement entre les deux dispositifs. Cela consiste à trouver la matrice de rotation X permettant de déduire les rotations de la caméra B à partir des rotations du capteur inertiel A (Fig. 2). Formellement, l’équation s’écrit :

$$AX = XB \quad (1)$$

La calibration hand-eye consiste à trouver la matrice X à partir de cette équation. On effectue plusieurs mesures de A et B (trois au minimum), obtenues pour différentes orientations du système caméra/capteur. De nombreuses méthodes existent pour résoudre un tel système, comme celle de Park et Martin [12] que nous avons utilisée.

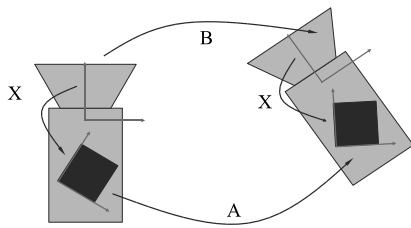


FIG. 2 – Transformation caméra/capteur.

3.2 Stratégie d’intégration

Notre stratégie pour intégrer les deux systèmes repose sur le fait que le suivi basé vision donne généralement une bonne estimée de la position de la caméra, sauf en cas de mouvement brusque de l’utilisateur. Le capteur est alors utilisé pour prédire la position des primitives à suivre lorsqu’un tel cas ce produit. En ce sens, il n’y a donc pas fusion de l’information issue des deux capteurs, mais utilisation de l’aide pouvant être apportée par le capteur en cas d’échec du suivi basé vision.

Nous n’utilisons pas systématiquement le capteur car sa faible précision peut nuire considérablement à la qualité ou à la robustesse du suivi. L’exemple suivant illustre ce point : la limite de confiance de 99% est atteinte sur l’axe z du capteur pour une erreur de $2.58 \times \sigma_z = 1.29^\circ$. Cela correspond à une erreur de 23 pixels horizontaux au centre de l’image, pour une distance focale classique de 1024 pixels. Cette amplitude d’erreur peut avoir plusieurs effets, selon la manière dont le capteur est utilisé : si les données capteurs sont systématiquement prises en compte pour le calcul de la rotation de la caméra, l’augmentation sera alors très peu précise et peu stable ; si les données capteur sont utilisées pour prédire la position des points d’intérêt à partir de leur position dans l’image précédente, le suivi échouera dans 1% des cas pour une fenêtre de recherche de 20 pixels, soit en moyenne toutes les 5 secondes pour une séquence de 20 images par seconde.

Une autre raison pour laquelle nous avons choisi de ne pas utiliser le capteur systématiquement est de ne pas devoir synchroniser la caméra et le capteur pour chaque image. En effet, une synchronisation précise doit être réalisée entre les deux systèmes, notamment lors de mouvements brusques [6]. Par conséquent, une mauvaise synchronisation aurait comme effet d’associer un mouvement abrupt à une mauvaise image, et par conséquent de rendre le suivi impossible. De plus, les expérimentations effectuées dans la partie 4 montrent que le délai de synchronisation n’est pas constant dans le temps et doit être estimé en temps réel, ce qui est couteux en termes de temps de calculs et ralentit le suivi. La contrainte temps réel de la RA nous conduit donc à utiliser le capteur seulement lorsque cela est nécessaire.

3.3 Prédiction capteur

Nous allons étudier dans cette partie la façon dont est intégré le capteur inertiel pour guider la phase d’appariement des points d’intérêt après un mouvement brusque de la caméra. La méthode peut être appliquée à tout type de mouvement détecté par un capteur, en ayant pris soin de mesurer la précision de ce dernier.

Lors d’un mouvement de rotation de la caméra, l’image entière est transformée selon une matrice H de dimension 3×3 appelée *matrice d’homographie*, et donnée par [13] :

$$H = KBK^{-1}, \quad (2)$$

où K est la matrice des paramètres intrinsèques de la caméra et B sa matrice de rotation. Cela signifie qu’un pixel à

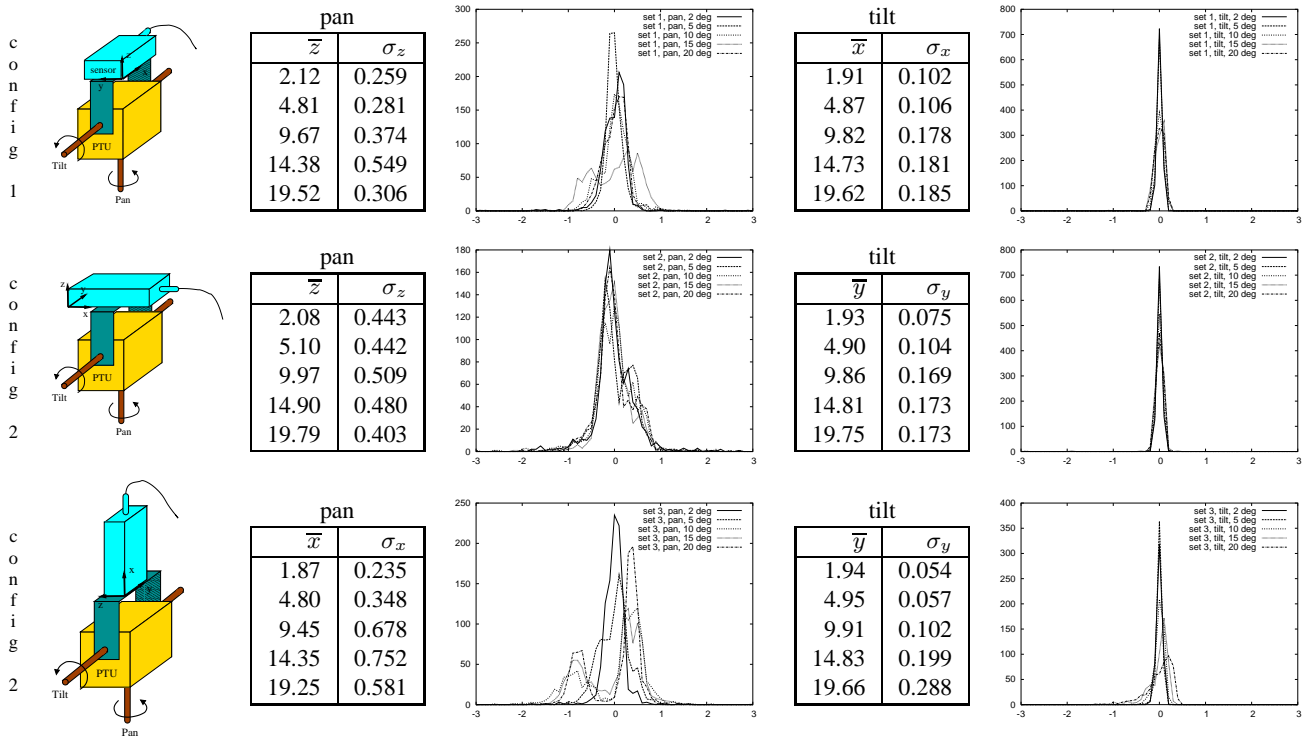


FIG. 1 – Mesures de précision du capteur inertiel XSens MT9-B sur ses trois axes de rotation.

la position m dans l'image précédente apparaît à la position $m' = Hm$ dans l'image courante (m et m' sont exprimés en coordonnées homogènes). Par conséquent, on déduit des équations (1) et (2) une fonction p prédisant la nouvelle position d'un pixel m , sachant que la matrice de rotation du capteur A est connue :

$$p(m) = KX^tAXK^{-1}m. \quad (3)$$

3.4 Propagation de l'erreur

Le capteur est non seulement utilisé pour la prédiction de la position des points d'intérêt, mais aussi pour définir des zones de recherche optimales. Ces zones de recherche sont des ellipses obtenues par propagation des erreurs capteur lors de la phase de calibration et d'acquisition des angles d'Euler. Ces erreurs peuvent être prises en compte dans une matrice de covariance sur les angles $\alpha_A, \beta_A, \gamma_A$:

$$\Sigma_A = \begin{pmatrix} \sigma_z^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_x^2 \end{pmatrix},$$

(les covariances inter-axes se sont avérées négligeables dans nos expérimentations). Cette matrice nous permet d'obtenir la matrice de covariance Σ_X de la matrice de rotation hand-eye, en utilisant les calculs de propagation d'erreur présentés en annexe. Pour une position prédite $m' = p(m)$, nous pouvons alors calculer, en utilisant l'approximation linéaire suivante, une matrice de covariance

$\Sigma_{m'}$:

$$\Sigma_{m'} = J_{X/m}\Sigma_XJ_{X/m}^t + J_{A/m}\Sigma_AJ_{A/m}^t,$$

où

$$J_{A/m} = \begin{pmatrix} \frac{\partial p}{\partial \alpha_A} & \frac{\partial p}{\partial \beta_A} & \frac{\partial p}{\partial \gamma_A} \end{pmatrix},$$

$$J_{X/m} = \begin{pmatrix} \frac{\partial p}{\partial \alpha_X} & \frac{\partial p}{\partial \beta_X} & \frac{\partial p}{\partial \gamma_X} \end{pmatrix}$$

(les paramètres intrinsèques de la caméra sont supposés fixes et pré-calculés). On en déduit donc une ellipse de confiance autour de m' , définie par :

$$X^t\Sigma_{m'}^{-1}X \leq 9.21, \quad (4)$$

où la valeur 9.21 correspond à la limite de confiance de 99% pour une loi du χ^2 à deux degrés de liberté.

4 Synchronisation en ligne

Afin de déterminer le délai de synchronisation entre la caméra et le capteur, nous avons dans un premier temps développé une méthode de détection de mouvements brusques. Des rotations abruptes sont effectuées dans une séquence d'images, et les changements brusques générés dans l'image sont détectés à l'aide d'un critère proposé par Prager [14] dans le cadre de la calibration temporelle d'une sonde à ultra-sons. Ces changements image sont mis en relation avec les rotations brusques du capteur. On observe alors que le délai d'acquisition des données capteur et des données caméra n'est pas constant au cours du temps

(voir la figure 3 où trois mouvements abrupts sont effectués, mettant en évidence des délais de synchronisation très variables). La synchronisation entre le capteur et la caméra doit donc être effectuée en ligne, à chaque fois que la capteur a besoin d'être utilisé.

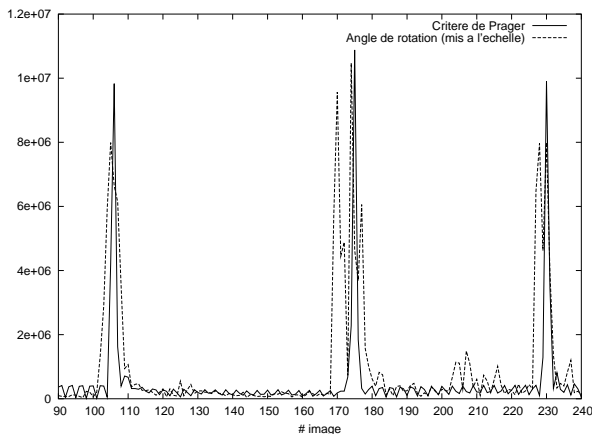


FIG. 3 – Le délai d’acquisition des données entre le capteur inertielle et la caméra n’est pas constant dans le temps. Les pics en trait plein correspondent aux changements brusques dans l’image et les pics en pointillés sont les rotations brusques du capteur. Dans cet exemple où 3 rotations ont été effectuées, le délai d’acquisition entre le capteur et la caméra sont respectivement de 1, 5 et 2 images.

Dans notre situation, les données capteurs sont toujours disponibles avant les données images. En effet, le temps de capture d’une image (que nous appellerons *cycle d’acquisition*) est beaucoup plus important que le temps nécessaire pour acquérir une rotation fournie par le capteur inertielle. Pour chaque cycle d’acquisition, les rotations relatives fournies par le capteur sont donc stockées dans un buffer afin de calculer l’*homographie capteur*, qui correspond à la rotation complète du capteur lors d’un cycle d’acquisition. Ces homographies sont alors utilisées pour guider le suivi des points d’intérêt lorsque le système basé vision échoue. Notre méthode de synchronisation en ligne est illustrée en figure 4. Elle consiste en trois étapes :

1) identification d’un échec de suivi basé vision

Une façon d’évaluer un échec du système basé vision est de détecter une chute dans le nombre d’inliers (c’est-à-dire de primitives correctement appariées). Un tel échec est généralement dû au fait que les points d’intérêt se trouvent en dehors de la zone de recherche après un mouvement abrupt. Le nombre d’inliers est un critère particulièrement pertinent dans le cadre du suivi multi-planaire, car l’utilisation de la contrainte homographique permet d’obtenir un comptage très robuste [13]. Dans notre exemple, ce nombre chute fortement dans l’image 11, ce qui signifie que le capteur doit prendre le relai dans cette image.

2) identification des données capteur correspondantes

Lorsqu’un échec du système basé vision a été détecté, l’homographie capteur correspondante doit être trouvée. Pour cela, nous simulons la nouvelle position des points d’intérêt en utilisant les homographies capteur stockées dans le buffer (de la plus ancienne à la plus récente). La première homographie faisant “virtuellement” échouer le suivi est celle que l’on considère comme correspondant au mouvement brusque de la caméra.

En effet, dans le suivi basé vision, l’appariement échoue lorsque les points d’intérêt se trouvent en dehors des zones de recherche. Par conséquent, les données capteur recherchées sont sensées prédire des positions de points hors des zones de recherche. Notre procédure revient donc à choisir l’homographie qui fait chuter le nombre d’inliers *prédits* par le capteur (points propagés à l’intérieur des zones de recherche par l’homographie capteur). Dans notre exemple, cela nous permet d’identifier à la troisième position du buffer les données capteur correspondant à l’échec du suivi basé vision.

Remarquons qu’une méthode plus simple consisterait à détecter par seuillage les variations brusques des rotations capteur. Cependant, la valeur du seuil dépendrait alors de la position du système caméra/capteur par rapport à la scène, ce qui ne permettrait pas d’établir un critère objectif pour la synchronisation.

3) utilisation du capteur pour réinitialiser la phase d’appariement

Une fois que l’homographie capteur correspondant au mouvement brusque a été identifiée à la position p dans le buffer, les données du capteur inertielle sont utilisées pour réinitialiser la phase d’appariement. Pour cela, les homographies du buffer sont cumulées à partir de la position p , de manière à calculer la transformation complète correspondant à l’échec du système basé vision. Le cumul est nécessaire car il arrive souvent que le mouvement de rotation brusque soit plus long qu’un seul cycle d’acquisition.

5 Le système ARIS

Le but du projet ARIS est d’utiliser les avancées récentes de la RA pour le commerce électronique, où des produits en vente peuvent être présentés dans leur contexte futur d’utilisation. Une unité mobile de RA a été développée, où les produits virtuels 3D (par exemple des meubles) sont directement visualisés dans leur environnement réel, en prenant en compte la géométrie de la scène et l’éclairage global, ainsi que des aspects collaboratifs en ce qui concerne la visualisation.

Ce paragraphe présente la partie reconstruction et suivi de la scène, puis interaction avec l’environnement. Les expérimentations ont été effectuées sur une scène miniature (Fig. 5). Des vidéos sont aussi disponibles sur notre site internet¹

¹<http://www.loria.fr/isa/movies/rfia.html>

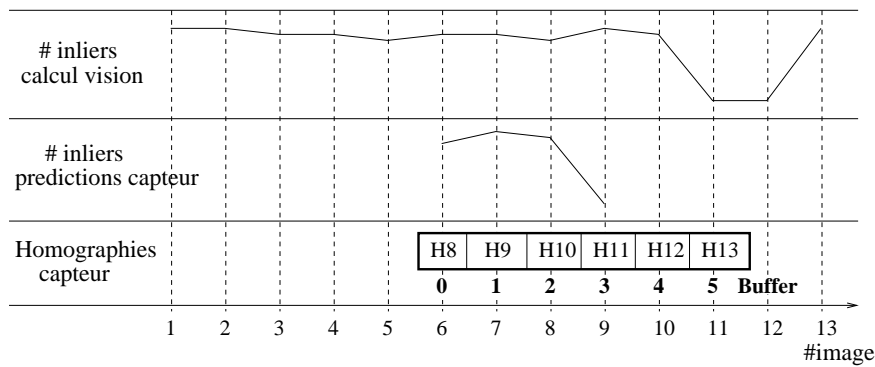


FIG. 4 – Illustration de la synchronisation en ligne entre la caméra et le capteur.

Suivi basé vision :

Notre système de suivi collaboratif repose sur le système basé vision présenté en [9] : des plans sont définis, et sur chacun, les points d'intérêt sont suivis et appariés. Ces appariements fournissent alors des homographies permettant de calculer la nouvelle position de la caméra (en supposant que les équations des plans soient connues). Les mauvais appariements sont identifiés au moyen de l'algorithme de RANSAC (Fig. 5.(a)), et les données capteurs sont utilisées comme précisé précédemment. La robustesse du système s'en trouve donc améliorée et l'utilisateur possède alors une plus grande liberté de mouvements. Le suivi est effectué à la vitesse de 15 images par seconde sur un ordinateur portable Pentium 3Ghz et avec une webcam classique. La calibration capteur/caméra est réalisée en filmant une mire de calibration selon plusieurs points de vue. Des marqueurs sont placés sur la mire afin d'automatiser le processus de calibration et de retrouver directement la position de la caméra (Fig. 5.(b)).

Reconstruction de la scène :

La reconstruction de la scène en 3D est utilisée à la fois pour le suivi, pour le rendu photo-réaliste, et pour les interactions entre les objets réels et virtuels. La géométrie de la scène est acquise à partir d'une seule image à l'aide d'un outil interactif de reconstruction (basé sur des primitives comme des parallélépipèdes rectangles redimensionnables...). Les paramètres intrinsèques de la caméra et le point de vue initial sont retrouvés à partir des sommets d'un poster posé dans un coin de la pièce (Fig. 5.(c)).

Interaction avec l'environnement :

Puisque les positions de la caméra sont calculées dans un repère global, des objets virtuels peuvent être ajoutés à tout instant dans la scène lors du suivi (Fig. 5.(d-f)). Ces objets sont placés à l'aide de la souris, et l'utilisateur peut les déplacer à sa guise dans la scène reconstruite. Les occultations et les collisions sont automatiquement gérées.

6 Résultats

6.1 Données synthétiques

Afin de vérifier et valider nos résultats concernant l'utilisation d'ellipses de confiance à la place de zones de recherche rectangulaires, nous avons simulé des centaines de calibrations caméra/capteur et de rotations capteur, en ajoutant un bruit blanc gaussien aux données capteur (voir la section 2).

Une bonne zone de recherche est une région qui maximise la probabilité $P(inlier)$ d'avoir le point d'intérêt correspondant à l'intérieur de la zone de recherche, minimisant le nombre de mauvais appariements $\#outliers$ dans cette région. Ces données ont été calculées en utilisant des zones de recherche rectangulaires de différentes tailles et des ellipses correspondant à diverses valeurs de confiance de l'équation (4). La figure 6 montre l'évolution du nombre d'outliers en fonction de $P(inlier)$, pour différents nombres de points d'intérêt n , choisis aléatoirement dans une fenêtre de taille 512×512 : les courbes correspondant aux zones elliptiques sont généralement en-dessous des courbes correspondant aux zones rectangulaires (moins d'outliers dans la zone de recherche pour une même probabilité de trouver l'homologue), avec des écarts significatifs. Par exemple, pour une zone rectangulaire de recherche de taille 20 pixels dans une fenêtre de 512×512 pixels, on trouve 12 outliers pour 2000 points. En utilisant des zones de recherche elliptiques, on obtient seulement 9 outliers (diminution de 25%) pour la même probabilité $P(inlier)$ de trouver le correspondant. Cette réduction est significative puisqu'elle diminue à la fois le risque d'ambiguïté et les temps de calculs utilisés pour l'appariement.

6.2 Séquences réelles

La figure 7.(a) montre le nombre de bons appariements obtenus avec la séquence miniature. Les périodes où le suivi échoue sont indiquées en traits pointillés. Elles correspondent à un mouvement brusque du système caméra/capteur. Le nombre d'inliers prédit par le capteur inertiel, en utilisant les homographies stockées dans le buffer pour l'image 173 sont montrées sur la figure 7.(b). On voit ici clairement l'avantage d'utiliser le critère retenu

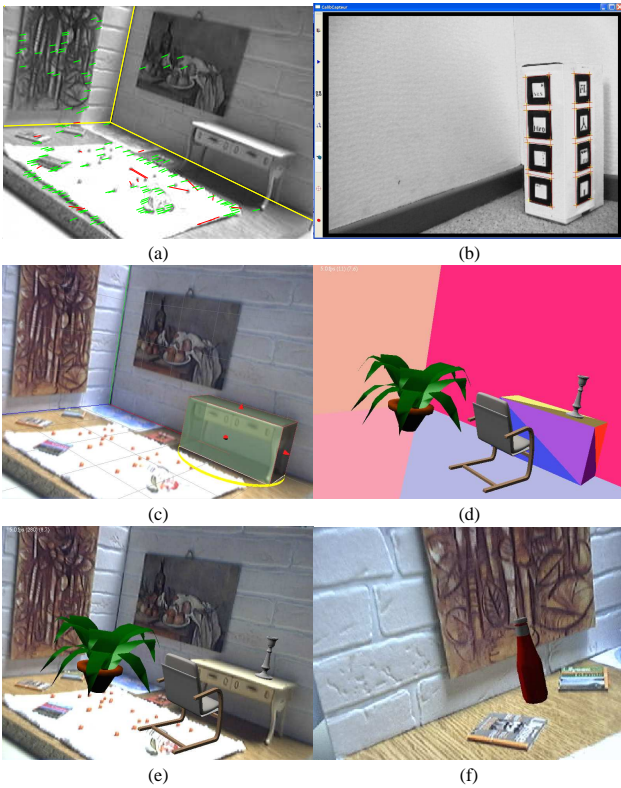


FIG. 5 – Suivi et interaction dans le système ARIS. (a) Suivi basé vision : les traits verts représentent les bons appariements entre l'image courante et l'image précédente. Les traits rouges représentent les mauvais appariements. (b) La calibration caméra/capteur est effectuée avec une mire de calibration utilisant des marqueurs. (c) Les paramètres de caméra et la géométrie de la scène sont retrouvés à partir d'une seule image. (d-f) Les objets virtuels sont placés dans la scène reconstruite, et l'utilisateur peut alors visualiser et interagir avec ces objets.

pour la synchronisation en ligne au lieu d'utiliser directement un seuillage sur les angles de rotations fournis par le capteur (Fig. 7.(c)).

Le suivi a aussi été réalisé sur des scènes réelles (les vidéos 3 et 4, disponibles sur notre site internet). La première a été filmée en intérieur, au sous-sol de notre laboratoire (Fig. 8). L'utilisateur se trouvait en conditions réelles, libre de ses mouvements, et les meubles virtuels sont insérés en temps réel. Au cours de cette expérimentation, on observe une dizaine de mouvements brusques où le capteur est mis à contribution. La seconde séquence a été prise en extérieur, à l'aide d'une caméra sur un trépied (Fig. 9) : dans cette configuration, l'image complète subit une homographie. Le capteur est utilisé 13 fois avec succès au cours de cette séquence.

7 Conclusion

Nous avons présenté une approche hybride pour la RA, basée sur la collaboration entre un capteur inertiel et un sys-

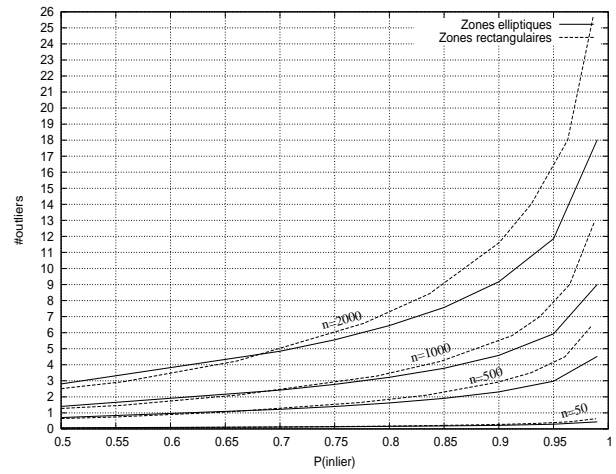


FIG. 6 – Pour différents nombres de points, nombre de mauvais appariements (outliers) attendus en fonction de la probabilité d'avoir un bon appariement (inliers) dans la zone de recherche (rectangle ou ellipse).

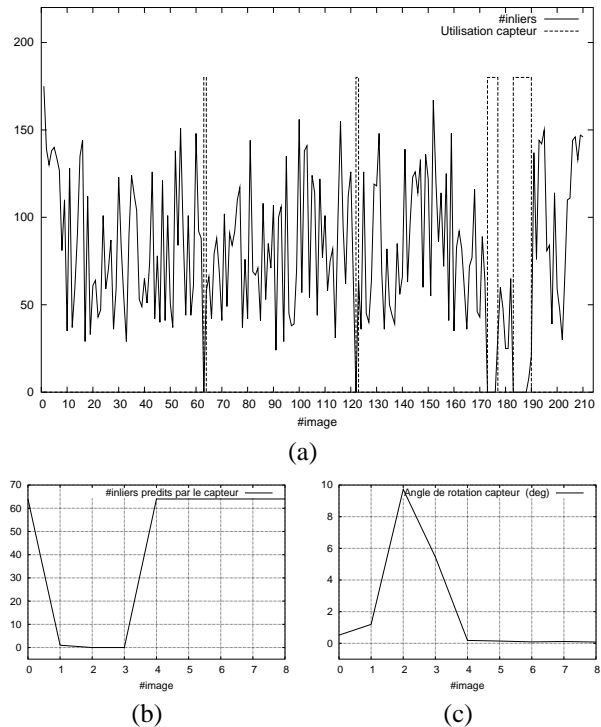


FIG. 7 – (a) Nombre de bons appariements obtenus dans la séquence miniature. (b),(c) Comparaison des deux critères de synchronisation sur l'image 173 de cette séquence.

tème de suivi basé vision. Des expériences dans des environnements non préparés ont été réalisées avec succès. Notre travail apporte donc des améliorations significatives pour les systèmes de RA en donnant à l'utilisateur final encore plus de libertés, sans qu'il ait à se soucier de la vitesse de ses mouvements. Comparé aux autres approches existantes, nous n'avons pas cherché à fusionner le cap-



FIG. 9 – Résultats obtenus en temps réel sur une séquence en extérieur. En haut : image 486 augmentée avec un cube virtuel. En bas : image 488 et les appariements lors du mouvement brusque sur la droite.

teur et la caméra pour chaque image acquise. Au contraire, le capteur est utilisé seulement lorsque cela s'avère nécessaire : le système basé vision est robuste et suffit pour effectuer le suivi lors de mouvements normaux. Si ce dernier échoue dans son suivi, le capteur prend le relai. Nous avons aussi pris en compte les erreurs capteurs afin d'optimiser les zones de recherche des points d'intérêt. Enfin, une synchronisation en ligne a été effectuée pour pallier le problème de temps de synchronisation non constant.

Plusieurs aspects de nos travaux peuvent cependant être améliorés : en effet, le capteur inertiel ne donne qu'une information sur la rotation : pour des mouvements quelconques, il arrive souvent qu'une large translation ait aussi lieu. L'ajout d'un capteur pouvant nous donner une information sur cette translation améliorerait encore plus significativement la robustesse du suivi. Ensuite, notre méthode basée vision peut en soi gagner en précision : en effet, pour de très nombreux algorithmes de suivi comme pour le notre, le calcul de la pose est séquentiel, et les approximations successives utilisées font parfois diverger le système.

On pourrait utiliser des marqueurs, ou encore des images clés pour détecter ces divergences et réinitialiser le suivi lorsque cela s'avère nécessaire.

Annexe : Propagation d'erreur lors de la calibration caméra/capteur

La matrice X de calibration caméra/capteur est donnée par :

$$A_i X = X B_i, \quad (5)$$

où les matrices (A_i, B_i) sont n matrices de rotation capteur/caméra. On cherche à calculer la matrice de covariance Σ_X des angles d'Euler $\alpha_X, \beta_X, \gamma_X$ de la matrice X . Les rotations de la caméra B_i sont supposées correctes alors que les rotations du capteur A_i contiennent des erreurs de mesure.

L'équation (5) peut être écrite $f(x, a) = \mathbf{0}$, où x est un vecteur de dimension 3 contenant les angles d'Euler de la rotation X , a est un vecteur de taille $3n$ contenant les angles d'Euler des rotations A_i , et $\mathbf{0}$ est le vecteur nul de dimension $9n$. Une approximation du premier ordre de $f(x, a)$ donne :

$$f(\bar{x}, \bar{a}) + \frac{\partial f}{\partial x}(\bar{x}, \bar{a}) (x - \bar{x}) + \frac{\partial f}{\partial a}(\bar{x}, \bar{a}) (a - \bar{a}) \approx \mathbf{0},$$

avec \bar{a} et \bar{x} les estimées de a et x . On en déduit :

$$C \Sigma_X C^t = D [\Sigma_A] D^t,$$

où $C = \frac{\partial f}{\partial x}(\bar{x}, \bar{a})$, $D = \frac{\partial f}{\partial a}(\bar{x}, \bar{a})$ et $[\Sigma_A]$ est une matrice $3n \times 3n$ $\begin{pmatrix} \Sigma_A & & 0 \\ & \Sigma_A & \\ 0 & & \Sigma_A \end{pmatrix}$. Par conséquent, Σ_X peut être exprimé comme :

$$\Sigma_X = (C^t C)^{-1} C^t D [\Sigma_A] D^t (C^t C)^{-1}.$$

Remerciements

Ces travaux ont été financés par l'Union Européenne, dans le cadre du projet ARIS (IST-2000-28707).

Références

- [1] R.T. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, et B. MacIntyre, Recent Advances in Augmented Reality, *IEEE Computer Graphics and Applications*, pp. 34-47, 2001.
- [2] T. Viéville, F. Romann, B. Hotz, H. Mathieu, M. Buffa, L. Robert, P. Facao, O. Faugeras, J.T. Audren, Autonomous navigation of a mobile robot using inertial and visual cues, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 1993
- [3] A. State, G. Hirota, D. Chen, W. Garrett, et M. Livingston, Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking, *Computer Graphics, Proceedings Siggraph New Orleans*, pp. 429-438, 1996.

- [4] W.A. Hoff, K. Nguyen, et T. Lyon, Computer vision-based registration techniques for augmented reality, *Proc. Conf. Intelligent Robots and Computer Vision*, pp.538-548, 1996.
- [5] K. Satoh, M. Anabuki, H. Yamamoto, et H. Tamura, A hybrid registration method for outdoor augmented reality, *International Symposium on Augmented Reality, Los Alamitos*, pp. 67-76, 2001.
- [6] B. Jiang, S. You, et U. Neumann, A robust hybrid tracking system for outdoor augmented reality, *IEEE Virtual Reality, Chicago*, pp. 3-10, 2004.
- [7] G. Klein, et T. Drummond, Tightly Integrated Sensor Fusion for Robust Vision Tracking, *Proceedings of the British Machine Vision Conference, Cardiff*, pp. 787-796, 2002.
- [8] J. Borenstein et L. Feng, Gyrodometry : A new method for combining data from gyros and odometry in mobile robots, *IEEE ICRA, Minneapolis, Minnesota*, pp. 423-428, 1996.
- [9] G. Simon et M.O. Berger Pose Estimation for planar structures, *IEEE Computer Graphics and Applications, special issue on Tracking*, pp. 46-53, 2002.
- [10] C. Harris, Tracking with rigid models, *Active Vision* Blake and Yuille, MIT Press, 1992, chap. 4
- [11] K.W. Chia, A.D. Cheok, et S.J.D. Prince, Online 6 DOF Augmented Reality registration from natural features, *Proceedings of International Symposium on Mixed and Augmented Reality, Darmstadt*, pp. 305-313, 2002.
- [12] F. Park, et B. Martin, Robot sensor calibration : solving $AX=XB$ on the Euclidean group, *IEEE Transactions on Robotics and Automation*, vol 5, no 10, pp. 717-721, 1994.
- [13] R.I. Hartley et A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN : 0521623049, 2000.
- [14] R.W. Prager, A.H. Gee, et L. Berman, Stradx : real-time acquisition and visualization of freehand 3D ultrasound report *CUED/FINFENG/TR319, Cambridge University Department of Engineering*, 1998.

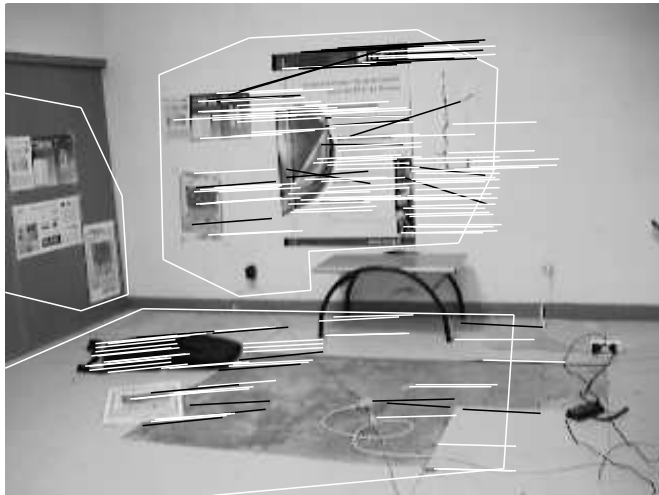


FIG. 8 – Résultats obtenus en temps réel à deux instants sur une séquence d'intérieur. En haut : images augmentées avant un mouvement brusque sur la droite. Au milieu : appariements obtenus en utilisant les prédictions du capteur (en blanc, les bons appariements (inliers), en noir les mauvais (outliers) identifiés par l'algorithme RANSAC). En bas : images augmentées après le mouvement brusque.