

Modélisation 3D d'objets par un capteur visuel déplacé par un opérateur

Benjamin Coudrin, Michel Devy, Ludovic Brèthes, Jean-José Orteu

► **To cite this version:**

Benjamin Coudrin, Michel Devy, Ludovic Brèthes, Jean-José Orteu. Modélisation 3D d'objets par un capteur visuel déplacé par un opérateur. ORASIS'09 - Congrès des jeunes chercheurs en vision par ordinateur, 2009, Trégastel, France, France. 2009. <inria-00404627>

HAL Id: inria-00404627

<https://hal.inria.fr/inria-00404627>

Submitted on 16 Jul 2009

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.

Modélisation 3D d'objets par un capteur visuel déplacé par un opérateur

3D Modelling of objects from a visual sensor moved manually by an operator.

Benjamin COUDRIN^{1,2} Michel DEVY¹ Ludovic BRETHERS² Jean-José ORTEU³

¹ CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France

Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France

² NOOMEO ; 10 av. de l'Europe, 31520 Ramonville Saint-Agne, France

³ Université de Toulouse, Ecole Mines Albi, CROMeP, Campus Jarlard, 81013 ALBI, France

mail : bcoudrin@laas.fr, michel@laas.fr, ludovic.brethes@noomeo.eu, jean-jose.orteu@enstimac.fr

Résumé

Nous évoquons dans cet article, la problématique de la modélisation 3D d'un objet par un capteur visuel déplacé manuellement par un opérateur. L'objet est posé sur un plan : l'opérateur déplace le capteur sur une demi-sphère centrée sur l'objet. Plusieurs fonctions doivent être développées pour construire un modèle par ce procédé : acquisitions 2D et 3D, estimation des déplacements du capteur à partir des points suivis dans la séquence d'images 2D et recalage des données 3D acquises en chaque point-de-vue 3D, construction du maillage et placage de texture. Cet article focalise sur le lien suivi-recalage lorsque le capteur est déplacé en continu, avec deux approches, recalages par ICP ou SLAM.

Mots Clef

Modélisation 3D, ajustement de faisceaux, estimation du déplacement, recalage, ICP, SLAM, suivi de points d'intérêt.

Abstract

This paper describes problems to be solved when modelling a 3D object from images acquired by a visual sensor moved manually by an operator. The object is set on a plane ; the sensor is moved on a semisphere centered on the object. In order to build a 3D model from such a protocol, several functions must be developed : 2D and 3D data acquisition, motion estimation from interest points tracked on the sequence of 2D images, registration of 3D data acquired from every 3D view point, construction of a geometrical model and texture mapping. This paper focuses only on the relationships tracking-registration from continuous data acquisitions when moving the sensor around the object, considering two registration methods, ICP or SLAM.

Keywords

3D modelling, bundle adjustment, motion estimation, registration, ICP, SLAM, Interest point tracking.

1 Introduction

Le problème de la modélisation 3D à partir d'un capteur mobile a trait à l'acquisition de données 3D à partir de positions distinctes de ce capteur, à leur agencement cohérent avec la réalité de la scène à modéliser, puis à leur représentation sous une forme géométrique (points, arêtes, surfaces, ...) dépendante de l'exploitation prévue pour le modèle final.

La phase d'acquisition est une étape sensorielle permettant d'obtenir une représentation partielle et bruitée de la scène réelle à modéliser. On exploite généralement des méthodes basées sur des capteurs télémétriques lasers ou vision (avec ou sans projection de lumière). Dans notre cas, il s'agit de travailler sur le capteur OptiNumTM de la société NOOMEOTM, capteur basé vision doté d'un système opto-électronique de numérisation qui permet l'acquisition de la géométrie et de la texture d'une scène sous la forme de nuages de points 3D. Ce capteur, sommairement décrit en section 2, est conçu pour être déplacé à la main.

Une acquisition 3D constitue donc une représentation parcellaire de l'espace qu'il faudra combiner avec d'autres acquisitions, prises sous d'autres points de vue. Fournir une modélisation réaliste passe donc par une étape de recalage de ces vues décorrelées. Cela impose de déterminer la position du capteur dans le repère monde au moment de chaque prise de vue. Cette position peut être déterminée facilement si l'on dispose d'informations proprioceptives, dans un contexte robotique par exemple [10]. Dans notre cas, le capteur étant porté à la main, nous ne disposons pas d'estimée a priori de sa position ; nous avons donc équipé OptiNumTM d'une caméra et d'une centrale inertielle. L'exploitation de ces capteurs pour estimer l'attitude du capteur, est décrite en section 3.

A partir de cette estimée, les tâches de localisation du capteur et de recalage des nuages de points 3D seront à terme traitées par l'une ou l'autre des méthodes suivantes. La première, décrite en section 4, se fonde sur des techniques d'optimisation. Très classiquement exploitée pour le pro-

blème du recalage, la méthode ICP [2, 3] optimise la transformation rigide entre deux modèles 3D en minimisant un critère SSD entre les points des modèles. Nous détaillerons cette approche ainsi que l'exploitation de données visuelles et inertielles pour améliorer et accélérer la convergence de l'algorithme [8].

La seconde, décrite en section 5, exploite le filtrage stochastique pour l'estimation de mouvement du capteur. Il s'agit de localiser le système en associant les données inertielles et des amers visuels détectés dans les vues précédentes, puis de les confronter aux données courantes. Cette approche est répandue dans les problèmes de navigation de robots mobiles et est connue sous le nom de SLAM (Simultaneous Localization and Mapping) [7].

Les autres étapes de la modélisation ne seront pas détaillées dans cet article. Elles seront seulement rappelées en section 6, dans nos futurs travaux.

2 Le capteur

Comme nous l'avons vu, le capteur OptiNum™ est un système de numérisation porté à la main développé par la société NOOME™. Basé vision, il permet, à chaque prise de vue, l'acquisition 3D dense de la scène dans le champ. Doté d'une caméra de localisation et d'une centrale intertielle, il est optimisé pour la numérisation d'objets contenus dans $1m^3$. Comme le montre la figure 1 OptiNum™ se présente sous la forme d'un pistolet muni d'un simple bouton poussoir utilisé pour le déclenchement de la numérisation. Deux pointeurs lasers ont été positionnés dans le boîtier pour converger à la distance de travail optimale du capteur.



FIGURE 1 – OptiNum™

En fonctionnement, l'appareil de capture est positionné face à l'objet. Le déclencheur est alors enfoncé, ce qui provoque l'allumage des pointeurs lasers et l'affichage sur écran d'un retour visuel pour aider l'utilisateur à ajuster la distance. Le relâchement du déclencheur provoque l'acquisition d'une image 3D. Le mécanisme de déclenchement peut être du type « one-shot », ou bien de type rafale, dans lequel plusieurs prises de vue sont effectuées successivement à intervalles de temps prédéterminés. Pour ces deux modes, il est possible d'effectuer une acquisition 2D par la caméra de localisation entre chaque prise d'image 3D

et ainsi de réaliser un suivi de points d'intérêt. On dispose également à chaque instant des données inertielles qui permettent de connaître l'attitude du capteur. C'est l'ensemble de ces données qui sont exploitées ici pour permettre au numériseur de générer un ensemble de vues 3D recalées.

3 Estimation de la position du capteur par mesures inertielles

Une centrale inertielle est un ensemble de capteurs permettant de mesurer le mouvement et la position d'un système dans l'espace. Cela inclut typiquement des mesures d'accélération et de vitesse angulaire ainsi qu'une boussole 3D et éventuellement un capteur GPS.

Nous avons intégré au système une centrale du commerce. Il s'agit de la centrale XSens MTi. Elle intègre un accéléromètre, un gyromètre et un magnétomètre trois axes, ainsi que des fonctionnalités de filtrage afin d'estimer et recalculer les biais induits par la température et d'assurer un calcul de pose.

Cette centrale exploite des technologies MEMS bas coût, permettant un faible encombrement mais souffrant particulièrement du bruit. Cela va contraindre fortement les possibilités de son exploitation dans un système de localisation. En effet, déduire les informations statiques (position, orientation) des mesures dynamiques (accélération, vitesse angulaire) nécessite l'intégration de ces mesures dans le temps. On s'expose à une dérive additive du fait de l'intégration de la composante de bruit en plus de l'intégration de l'accélération réelle mesurée.

L'exploitation de la troisième mesure (champ magnétique) permet cependant de calculer l'orientation du système à chaque instant en composant mesure magnétique et accélérométrie. En effet, la mesure magnétique est statique, et l'accéléromètre, comme tout corps sur terre, est soumis à l'accélération due à la gravité, mesurable. De ce fait, le vecteur d'accélération mesuré s'exprime :

$$\vec{g}_k = R_k \cdot \vec{g}_0$$

où $\vec{g}_0 = [0 \ 0 \ g]^T$ est le vecteur d'accélération dans le repère monde, et R_k est la matrice de rotation associée à l'attitude actuelle du capteur, matrice à estimer.

De manière similaire on peut exprimer la mesure magnétique relativement à l'attitude du système :

$$l_k = R_k \cdot l_0$$

où l_0 est la mesure du champ magnétique lorsque le capteur est aligné avec le repère monde.

Notre problème est donc de déterminer la matrice R_k qui minimise le critère suivant :

$$S(R_k) = \left\| x_k - \begin{bmatrix} g_k \\ l_k \end{bmatrix} \right\|^2$$

où $x_k = [a_x \ a_y \ a_z \ m_x \ m_y \ m_z]^T$ est le vecteur composé des mesures accélérométriques et magnétométriques à l'instant courant.

Afin de rendre plus robuste l'estimation il est possible de réaliser la minimisation en tenant compte des mesures précédentes. Il s'agit alors de trouver un compromis entre robustesse et réactivité.

En observant les variations du résidu il est possible de détecter les incohérences entre les sources de mesures et donc les perturbations de l'estimation.

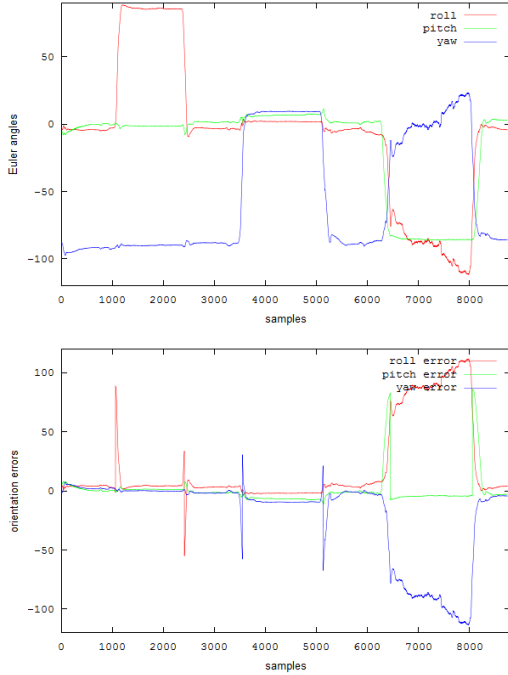


FIGURE 2 – Mesures d'attitude

La figure 2 montre l'estimation d'attitude du capteur et l'erreur associée. On a réalisé trois rotations sur un seul axe à la fois, puis un retour à la position initiale. La troisième rotation a été soumise à des vibrations fortes, on constate que le système n'est pas en mesure d'estimer l'attitude de manière fiable durant le temps de la perturbation mais retrouve rapidement une estimation juste lorsque celle-ci s'arrête.

On constate également que, sur les précédents mouvements, le système a besoin de quelques échantillons pour converger, provoquant des pics d'erreur. Cela est lié au fait que l'on a réalisé l'estimation en exploitant des mesures précédentes afin de rendre l'estimée plus robuste.

Exploiter les mesures inertielles pour déterminer le mouvement rectiligne est possible à condition de les coupler à d'autres mesures permettant de recalibrer les dérives. Nous utilisons ce type de méthodes dans nos approches de recalage décrites dans les sections suivantes.

Le capteur est couplé à un système de caméra. Nous calibrons la transformation entre les repères de ces deux capteurs à l'aide de la méthode de Mirzaei and al. [9].

4 Recalage par des techniques d'optimisation

L'algorithme ICP (Iterative Closest Points) a été introduit par Chen et Medioni [3] et Besl et McKay [2] ; c'est la méthode courante de recalage entre deux nuages de points 3D. Elle nécessite de fournir une estimée initiale de la transformation rigide entre ces deux nuages et l'affine itérativement en minimisant la somme des distances au carré entre un point et son plus proche voisin dans l'autre modèle. De nombreuses variantes ont depuis été proposées [11] afin d'améliorer la convergence de l'algorithme.

Les résultats de l'algorithme ICP dépendent fortement de la qualité de l'estimée initiale fournie. Celle-ci peut être déterminée en estimant la position spatiale du capteur, en associant des descripteurs de surface [6], en exploitant des entrées de l'opérateur. . .

L'exploitation des données inertielles permet d'obtenir une estimée de l'orientation du capteur au moment de la prise de vue et donc de la rotation à appliquer au nuage de points 3D à recalibrer. Dans ce cas, une paire de points 3D suffit pour estimer la composante translation de la transformation rigide entre les deux points de vue. En exploitant un plus grand nombre de paires, il est possible aussi de redéterminer l'orientation afin de rendre plus robuste l'estimée initiale.

La séquence d'images 2D obtenue entre les deux acquisitions 3D faites aux instants t_k et t_{k+l} , est exploitée pour réaliser le suivi de N points d'intérêt, et donc une mise en correspondance entre N points 3D. A l'instant t_k , le point d'intérêt q_i^k , est associé à un point 3D p_i^k ; il est suivi sur les l images suivantes ; à l'instant t_{k+l} , sa position q_i^{k+l} est associée à un point 3D p_i^{k+l} . On en déduit l'appariement 3D (p_i^k, p_i^{k+l}) .

On dispose donc de plusieurs méthodes pour produire l'estimée initiale. Sans la centrale inertielle, une méthode directe n'exploite que le résultat du suivi de N points [1] ($N > 5$). En utilisant les données inertielles, on peut exploiter une méthode directe avec 1 seul point pour trouver la translation, la rotation étant déterminée par les données inertielles. Enfin on peut réaliser une initialisation par une méthode directe à 1 point et exploiter d'autres appariements pour réaliser une optimisation.

On dispose de N paires de points 3D (p_i^k, p_i^{k+l}) mis en correspondance par suivi et une première estimation de la rotation R_0 entre les deux points de vue, obtenue par les mesures inertielles. En utilisant un des N appariements, on réalise une estimation de la translation t_0 entre les vues. Déterminer (R, t) la transformation rigide initiale à partir de ces données revient à minimiser le critère suivant sur les $N - 1$ points restants :

$$S(R, t) = \sum_{i=1}^{N-1} (p_i^k - (Rp_i^{k+l} + t))^2 \quad (1)$$

On minimise cette fonction en utilisant l'algorithme de

Levenberg-Marquardt que l'on initialise avec (R_0, t_0) .

4.1 Suivi de points

Le problème du suivi de points perçus depuis une caméra mobile, n'est pas un problème trivial. Le suivi est complexe dans le sens où l'on ignore les caractéristiques des mouvements de l'environnement et de la caméra, ainsi que la géométrie de la scène observée. De plus le travail sur les points ne permet d'exploiter que des caractéristiques locales. Dans notre cas, l'environnement est statique.

Les deux méthodes les plus courantes pour assurer un suivi robuste sont basées l'une sur l'algorithme de Kanade-Lucas-Tomasi [12] et l'autre sur le filtrage stochastique.

La méthode de Kanade-Lucas-Tomasi cherche à minimiser un critère de corrélation en exploitant l'hypothèse de conservation de luminance au voisinage du point. En considérant un point \mathbf{x} de l'image $I(\mathbf{x}, t)$ acquise à l'instant t , cette hypothèse peut être formalisée de la façon suivante :

$$I(\mathbf{x}, t) = I(\delta(\mathbf{x}), t + \tau) \quad \forall(\mathbf{x}) \in W \quad (2)$$

où W est le voisinage du point considéré et $\delta(\mathbf{x})$ exprime la transformation linéaire assurant le passage d'une image acquise au temps t à celle acquise au temps $t + \tau$. Cette transformation est supposée être une déformation affine :

$$\delta(\mathbf{x}) = A\mathbf{x} + \mathbf{d}$$

où \mathbf{d} est la translation du centre de la fenêtre et

$$A = \mathbf{1} + \begin{bmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{bmatrix}$$

avec $\mathbf{1}$ la matrice identité.

Du fait du bruit et de l'imperfection du modèle de mouvement, on ne peut satisfaire l'équation (2), A et \mathbf{d} sont donc déterminés par la minimisation d'une mesure de similarité :

$$\epsilon = \sum_W (I(\delta(\mathbf{x}), t + \tau) - I(\mathbf{x}, t))^2 \quad (3)$$

Considérant que les déplacements sont faibles, donc que l'acquisition des images est suffisamment rapide, on peut approximer $I(\delta(\mathbf{x}), t + \tau)$ par un développement en série de Taylor au premier ordre. En remplaçant son expression dans l'équation (3) et en différenciant par rapport aux inconnues D et \mathbf{d} , on obtient le système linéaire suivant :

$$T\mathbf{z} = \mathbf{a} \quad (4)$$

avec $\mathbf{z} = [d_{xx} \ d_{yx} \ d_{xy} \ d_{yy} \ d_x \ d_y]^T$, le vecteur collectant les inconnues de la déformation D et du déplacement \mathbf{d} et, en notant I_t, I_x, I_y les gradients temporels et spatiaux calculés au point \mathbf{x} , T et \mathbf{a} sont donnés par :

$$\mathbf{a} = -\tau \sum_W I_t [xI_x \ xI_y \ yI_x \ yI_y \ I_x \ I_y]^T$$

$$T = \sum_W \begin{bmatrix} U & V \\ V^T & G \end{bmatrix}$$

$$U = \begin{bmatrix} x^2 I_x^2 & x^2 I_x I_y & xy I_x^2 & xy I_x I_y \\ x^2 I_x I_y & x^2 I_y^2 & xy I_x I_y & xy I_y^2 \\ xy I_x^2 & xy I_x I_y & y^2 I_x^2 & y^2 I_x I_y \\ xy I_x I_y & xy I_y^2 & y^2 I_x I_y & y^2 I_y^2 \end{bmatrix}$$

$$V^T = \begin{bmatrix} xI_x^2 & xI_x I_y & yI_x^2 & yI_x I_y \\ xI_x I_y & xI_y^2 & yI_x I_y & yI_y^2 \end{bmatrix}$$

$$G = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Dans l'équation (4), \mathbf{z} est obtenu en utilisant un schéma itératif de Newton-Raphson. Le modèle affine est utilisé pour comparer le point considéré et son voisinage entre la première image et l'image courante afin de jauger la qualité du tracking.

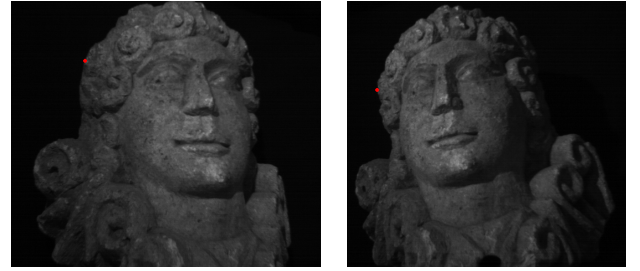


FIGURE 3 – Exemple d'erreur d'occultation

La figure 3 illustre un cas d'occultation du point, les images 300 et 369 sont représentées : avec le changement du point de vue, le voisinage du point devient très différent entre les deux images et la corrélation se dégrade fortement.

Cette méthode peut être encore améliorée en utilisant le filtrage stochastique pour estimer le mouvement apparent de chaque point suivi dans la séquence d'images. Suivre un point au long d'une séquence revient à estimer l'état caché $\mathbf{x}_{0:n} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n\}$ d'un processus Markovien, d'équation de transition $p(\mathbf{x}_k | \mathbf{x}_{k-1})$, conditionnellement à l'ensemble $\mathbf{z}_{1:n} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n\}$ des mesures de l'état. Par conséquent, on cherche à déterminer une approximation de la densité de probabilité a posteriori $p(\mathbf{x}_k | \mathbf{z}_{1:k})$. Le filtrage résoud ce problème récursivement en réalisant une prédiction de l'état en supposant $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1})$ connue :

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}$$

puis en utilisant la nouvelle mesure \mathbf{z}_k pour mettre à jour la prédiction :

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) d\mathbf{x}_k}$$

Le filtre de Kalman permet d'obtenir la solution optimale à ce problème dans le cas de modèles linéaires gaussiens. Le modèle que nous mettons en œuvre est formalisé par les relations suivantes :

$$\mathbf{x}_k = \Phi_k \mathbf{x}_{k-1} + \Gamma_k \mathbf{w}_k \quad (5)$$

$$\mathbf{z}_k = H_k \mathbf{x}_k + \mathbf{v}_k \quad (6)$$

où w_k et v_k sont des bruits blancs gaussiens indépendants de moyennes nulles et de covariances respectives Q_k et R_k , et $\mathbf{x}_k = [\Xi \ V]^T$ où Ξ représente la position du point dans l'image et V représente sa vitesse. On note \mathbf{P}_k la covariance associée à l'erreur d'estimation de l'état.

A l'instant t_k nous propageons l'estimation du pas précédent afin de produire une extrapolation *a priori* de l'état et de sa covariance, notés $\hat{\mathbf{x}}_k^-$ et \mathbf{P}_k^- :

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \Phi_k \hat{\mathbf{x}}_{k-1} \\ \mathbf{P}_k^- &= \Phi_k \mathbf{P}_{k-1} \Phi_k^T + Q_k \end{aligned}$$

On calcule le gain de Kalman, permettant de quantifier la croyance relative entre la mesure \mathbf{z}_k et la prédiction de l'état $\hat{\mathbf{x}}_k^-$:

$$K_k = \mathbf{P}_k^- H_k^T (H_k \mathbf{P}_k^- H_k^T + R_k)^{-1}$$

On réalise alors la mise à jour des prédictions pour obtenir l'estimation *a posteriori* de l'état et sa covariance associée :

$$\begin{aligned} \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - H_k \hat{\mathbf{x}}_k^-) \\ \mathbf{P}_k &= (I - K_k H_k) \mathbf{P}_k^- \end{aligned}$$

4.2 Recalage ICP

En supposant que l'on dispose, grâce au suivi de points entre deux acquisitions 3D, d'une bonne estimée initiale de la transformation rigide entre deux vues 3D V_k et V_{k+l} , l'algorithme ICP va chercher à minimiser un critère de distance inter-points pour rapprocher les modèles.

Il s'agit d'un algorithme itératif utilisant un ensemble de points $\{p_i^k\}$ sélectionnés dans la vue V_k . A chaque itération on va chercher à :

- appairer les points p_i^k avec leur plus proche voisin dans la vue V_{k+l} . On utilise pour cela une structure *k-d tree*,
- pondérer et rejeter des appariements,
- estimer la transformation rigide par minimisation d'un critère SSD sur les points appariés.

On applique alors la transformation à la vue V_{k+l} pour la rapprocher de V_k . On itère à nouveau jusqu'à satisfaction d'un critère de convergence de distance entre les vues ou en fixant une limite sur le nombre d'itérations.

La sélection des appariements se fait au moyen de deux filtres. Le premier assure l'unicité d'appariement [10]. Un point de V_{k+l} ne peut être apparié qu'à un seul point de V_k . La phase d'appariement produit une liste de candidats dans V_{k+l} pour chaque point p_i^k de V_k . Ces listes sont ensuite parcourues afin de garantir à la fois l'unicité des appariements et le choix des paires optimales selon la distance 3D. Le second filtre est un critère statistique basé sur la distribution des distances entre les points appariés [14]. On met à jour itérativement un seuil D_{max}^{k-1} et on supprime les appariements présentant une distance supérieure à ce seuil.

La valeur initiale D_{max}^0 est fixée à une valeur élevée. On modifie le seuil en fonction de la moyenne μ des distances et de l'écart-type σ :

- si $\mu < D_{max}^{k-1}$ alors $D_{max}^k = \mu + 3\sigma$,
- sinon si $\mu < 3D_{max}^{k-1}$ alors $D_{max}^k = \mu + 2\sigma$,
- sinon si $\mu < 6D_{max}^{k-1}$ alors $D_{max}^k = \mu + \sigma$,
- sinon $D_{max}^k = \xi$.

ξ est la valeur médiane sur toutes les distances.

Nous disposons donc d'un ensemble de N paires de points (p_i^k, p_i^{k+l}) . Les appariements n'étant pas parfaits, estimer la transformation rigide entre ces deux ensembles revient à minimiser une mesure de similarité :

$$\epsilon = \sum_{i=1}^N [p_i^k - (Rp_i^{k+l} + t)]^2 \quad (7)$$

Arun et al. [1] proposent une méthode de minimisation au sens des moindres carrés basée sur la décomposition SVD d'une matrice représentant l'ensemble des appariements.

La méthode propose de résoudre le problème en deux temps, en séparant la rotation et la translation. On peut se rendre indépendant à la translation en exprimant chaque point par rapport au centre de masse du nuage dont il est issu.

$$\begin{aligned} p_m^k &= \frac{1}{N} \sum_{i=1}^N p_i^k & p_m^{k+l} &= \frac{1}{N} \sum_{i=1}^N p_i^{k+l} \\ q_i^k &= p_i^k - p_m^k & q_i^{k+l} &= p_i^{k+l} - p_m^{k+l} \end{aligned}$$

Ainsi on peut exprimer l'équation (7) en fonction de R uniquement :

$$\epsilon = \sum_{i=1}^N [q_i^k + p_m^k - Rq_i^{k+l} - Rp_m^{k+l} - t]^2$$

Par construction, $p_m^k - (Rp_m^{k+l} + t) = 0$, donc on peut exprimer la quantité à minimiser par :

$$\epsilon = \sum_{i=1}^N [q_i^k - Rq_i^{k+l}]^2 \quad (8)$$

La méthode permet de déterminer R par minimisation du critère (8). t est obtenu ensuite par

$$t = p_m^k - Rp_m^{k+l}$$

On développe l'équation (8) :

$$\begin{aligned} \epsilon &= \sum_{i=1}^N (q_i^k - Rq_i^{k+l})^T (q_i^k - Rq_i^{k+l}) \\ &= \sum_{i=1}^N (q_i^{kT} q_i^k + q_i^{k+lT} R^T R q_i^{k+l} \\ &\quad - q_i^{kT} R q_i^{k+l} - q_i^{k+lT} R^T q_i^k) \\ &= \sum_{i=1}^N (q_i^{kT} q_i^k + q_i^{k+lT} q_i^{k+l} - 2q_i^{kT} R q_i^{k+l}) \end{aligned}$$

Par conséquent, minimiser ϵ est équivalent à maximiser

$$\begin{aligned} F &= \sum_{i=1}^N q_i^{kT} R q_i^{k+l} \\ &= \text{Trace} \left(\sum_{i=1}^N R q_i^{k+l} q_i^{kT} \right) = \text{Trace}(RH) \quad (9) \end{aligned}$$

avec

$$H = \sum_{i=1}^N q_i^{k+l} q_i^{kT}$$

Soit la décomposition SVD de H :

$$H = U\Lambda V^T$$

Dans cette égalité, les matrices U et V sont orthonormales et Λ est une matrice diagonale positive contenant les valeurs propres (ou singulières). Soit

$$X = VU^T \rightarrow XH = VU^T U\Lambda V^T = V\Lambda V^T$$

XH est symétrique positif. Or, pour toute matrice B ortho-normale 3×3 :

$$\text{Trace}(XH) \geq \text{Trace}(BXH)$$

Dans ces conditions, X maximise la fonction 9. Si $\det(X) = +1$, $X = R$. Si $\det(X) = -1$, X est une réflexion et deux cas se présentent :

- une des valeurs singulières de H est nulle. Dans ce cas, la rotation recherchée est :

$$X' = V'U^T$$

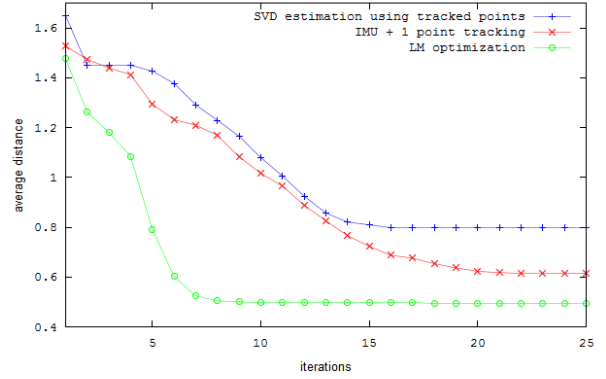
où V' est obtenu en changeant le signe de la troisième colonne de V .

- aucune des valeurs singulières de H n'est nulle. Dans ce cas, la méthode proposée ne convient pas, on préférera alors s'orienter vers des solutions de type RANSAC, par exemple.

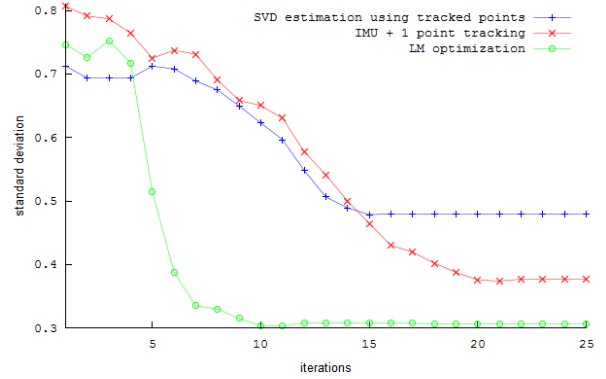
La figure 4 montre une comparaison de recalages ICP en faisant varier la méthode de production de l'estimée initiale. Les données utilisées pour ce test sont constituées de deux acquisitions 3D entre lesquelles ont enregistré une séquence d'images à 5Hz et une séquence de mesures inertielles à 100Hz. La séquence d'images est utilisée pour réaliser un suivi de points permettant d'obtenir une liste de points 3D appariés.

L'estimée initiale est produite selon les méthodes décrites plus haut :

- On utilise uniquement les paires de points pour estimer la transformation entre les nuages. (bleu)
- On utilise les données de la centrale pour estimer la rotation entre les deux acquisitions 3D et une paire de points 3D pour trouver la translation. (rouge)
- On utilise la méthode précédente pour initialiser une minimisation par la méthode de Levenberg-Marquardt.



(a)



(b)

FIGURE 4 – Résultats de recalage ICP utilisant les différentes méthodes d'estimée initiale. (a) Moyenne des distances entre points appariés. (b) Ecart type de la distribution des distances.

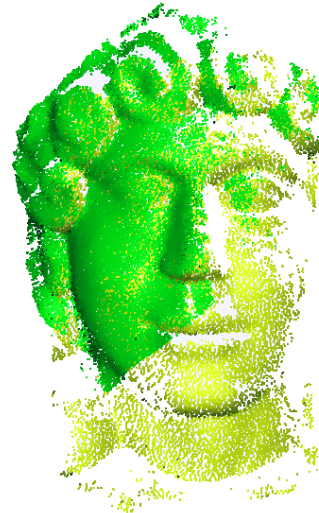


FIGURE 5 – Deux nuages de points recalés par ICP

On note que les deux premières méthodes, n'utilisant pas l'ensemble des informations disponibles produisent des es-

timées initiales de moins bonne qualité pour l'algorithme ICP. En effet, en utilisant la troisième méthode le nombre d'itérations de l'algorithme d'optimisation est réduit et l'erreur finale est réduite.

La figure 5 est le résultat du recalage de la séquence de test en utilisant la troisième méthode d'estimée initiale.

5 Recalage par une approche SLAM

Le SLAM (Simultaneous Localization and Mapping) est un processus par lequel un système construit une carte de son environnement et l'utilise pour s'y localiser. Il s'agit d'une des grandes thématiques de la robotique mobile et un grand nombre de solutions ont été proposées pour résoudre le problème. Typiquement on utilise un télémètre laser afin de localiser un robot sur une carte 2D. Afin de dépasser cette limitation, on a vu se développer des solutions SLAM basé vision [4], permettant d'exploiter un capteur à coût réduit. Les travaux récents sur le SLAM basé vision proposent des solutions temps-réel mono caméra.

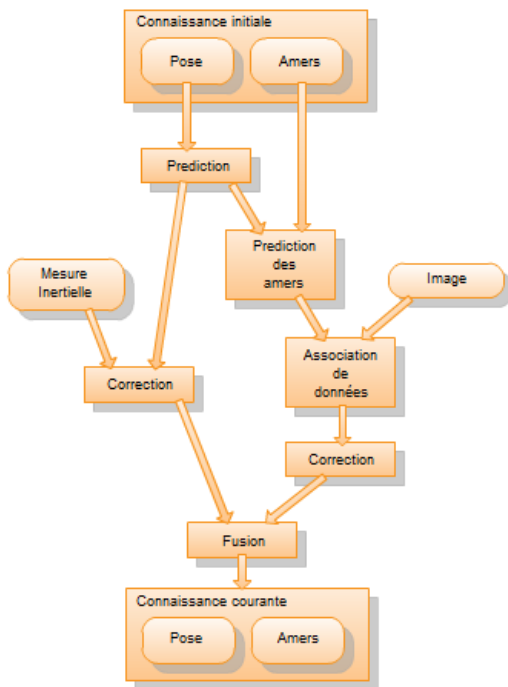


FIGURE 6 – Le processus SLAM basé vision et inertiel

Dans notre cas, nous utilisons une approche SLAM afin d'obtenir une estimation de la pose de la caméra. Cette tâche est effectuée en exploitant la caméra et la centrale inertielle.

L'initialisation de la procédure nécessite de fournir en entrée du système une connaissance préalable de la position de la caméra et un ensemble d'amers visuels. Pour définir les amers visuels, comme dans le cas du suivi décrit en

section 4.1, on réalise deux acquisitions successives 2D et 3D, en supposant un mouvement nul entre ces deux acquisitions.

A l'instant t_0 on acquiert un nuage de points 3D. Sur l'image 2D acquise ensuite, on lance une détection de points d'intérêt bien répartis dans l'image : on divise l'image en sous-images et on sélectionne un point par sous-image, en utilisant le critère de Shi-Tomasi [12] par exemple. Comme le mouvement entre les deux acquisitions est supposé nul, on a une association d'un point d'intérêt (u_i, v_i) détecté dans l'image 2D, avec un point (X_i, Y_i, Z_i) obtenu dans le nuage 3D. Pour chaque amer visuel, on conserve donc sa position 3D (X_i, Y_i, Z_i) , son apparence obtenue dans un voisinage autour du point (u_i, v_i) dans l'image 2D, et afin de réaliser ultérieurement une détection d'occultations, une approximation locale de la normale $N_i = (\alpha_i, \beta_i)$, calculée dans un voisinage autour du point (X_i, Y_i, Z_i) dans le nuage 3D.

La carte stochastique du processus SLAM est un ensemble $(\mathbf{x}_k, \mathbf{P}_k)$ où $\mathbf{x}_k = [\hat{X} \ y_0 \ y_1 \ \dots \ y_n]^T$ est le vecteur d'état combinant l'état de la caméra et l'ensemble des états des amers détectés, et \mathbf{P}_k est la matrice de covariance associée détaillant la croyance que l'on a dans l'estimation \mathbf{x}_k .

L'état de la caméra donne une représentation de la pose du système. A l'instant t_0 , on suppose que la caméra est en $(0, 0, 0)$ avec une erreur nulle (ou très faible pour éviter les problèmes numériques) ; les amers sont initialisés comme décrit plus haut, l'incertitude initiale est liée à la précision nominale du capteur 3D (figure 7 gauche). On majore cependant cette incertitude afin de prendre en compte un micro-mouvement de l'opérateur entre l'acquisition 3D et l'acquisition 2D, qui ne sont pas garanties synchrones.

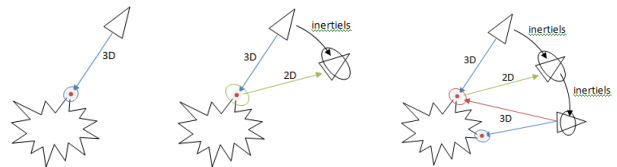


FIGURE 7 – Apport des différents types de mesures à la réduction de l'incertitude

Le problème SLAM est un problème de filtrage Bayésien. Plusieurs méthodes d'estimation ont été proposées pour le résoudre ; dans notre cas nous utilisons le Filtre de Kalman Étendu (EKF). Les figures 6 et 7 illustrent le processus.

L'état du capteur et son environnement est observé par trois capteurs : un capteur inertiel, une caméra 2D et un numériseur 3D. Chaque source d'information va permettre la mise à jour de tout ou partie de la carte SLAM.

On commence par réaliser une prédiction de l'état de la caméra (figure 7 centre). L'exploitation des mesures inertielles permet d'assurer la correction du mouvement et de

la position du capteur. On peut alors réaliser une prédiction sur la visibilité des amers par la caméra du capteur. On exploite l'information de normale en la comparant à la direction de l'axe optique afin de ne pas chercher à observer les amers supposés occultés.

En exploitant la position prédite de la caméra, on détermine la projection de chaque amer visible sur le plan rétinien de la caméra ainsi que l'incertitude de cette projection. En réalisant un parcours dans cette zone d'incertitude on va chercher une association entre cet amer avec un point image, par corrélation de leurs apparences. La déformation de fenêtre de corrélation peut être facilement déterminée en utilisant la pose relative entre la caméra et le point.

Si l'on arrive à corréler le point on obtient une observation. Chaque observation obtenue par recherche des amers dans l'image courante est exploitée afin de mettre à jour l'état et sa covariance.

Une acquisition 3D donnera une information directe sur les amers (figure 7 droite). Il nous faut tout d'abord utiliser l'estimation de la position de la caméra afin de recalibrer cette mesure 3D avec le repère monde puis on cherche à associer les amers de la carte avec un point du nouveau nuage 3D, à l'aide d'un *k-d tree* par exemple. L'incertitude apportée par la mesure est liée à la résolution du capteur 3D.

Ces trois types d'acquisitions fonctionnant à des fréquences différentes, il faut mettre en place une gestion de l'état permettant de garder trace d'instantanés précédents afin que chaque mesure puisse être exploitée selon un état précédent cohérent avec son modèle. L'étape de fusion permet d'aggréger les estimations et de fournir un état le plus précis possible au regard des informations disponibles.

L'incertitude sur les positions des amers et la position de la caméra va avoir tendance à croître. La réobservation d'amers anciens de la carte, ou « fermeture de boucle » va permettre de réduire de façon importante l'incertitude globale. Il est donc important que l'opérateur adopte une trajectoire, lors de la numérisation, permettant la réobservation d'amers anciens.

6 Conclusions

Nous avons dans cet article, décrit nos travaux actuels sur la modélisation 3D d'objets par le nouveau capteur OptiNum™ de la société NOOMEO™, capteur déplacé à la main par un opérateur, autour d'un objet à numériser. Nous avons présenté nos résultats actuels sur l'estimation de l'attitude du capteur et sur le recalage des données acquises depuis différents points de vue par des techniques classiques ICP. Nous présentons, de plus, le principe d'un recalage par approche SLAM.

Par la suite, après recalage et fusion des vues, nous créerons un maillage entre les points du nuage afin de représenter la surface de l'objet [5]. Un placage de texture permettra d'enrichir le modèle final avec des informations photométriques. Comme le capteur est manipulé manuellement, il faut retourner à l'opérateur, des indications sur la qualité

courante de la reconstruction. L'exploitation de méthodes de planification perceptuelle [10] et d'une IHM minimale permettra de le guider lors de l'acquisition pour choisir la prochaine bonne vue afin d'optimiser la qualité et la complétude du modèle final.

Références

- [1] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-d point sets. volume 9, 1987.
- [2] P. Besl and N. McKay. A method for registration of 3-d shapes. In *Trans. PAMI*, volume 14, 1992.
- [3] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proc. IEEE Conf. on Robotics and Automation*, 1991.
- [4] A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *ICCV*, 2003.
- [5] H. Delingette. Simplex meshes : a general representation for 3d shape reconstruction. Technical report, INRIA, Sophia Antipolis, France, 1994.
- [6] A. Johnson and M. Herbert. Surface registration by matching oriented points. In *Proc. 3DIM*, 1997.
- [7] J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *IROS*, 1991.
- [8] L.G.B. Mirisola, J. Lobo, and J. Dias. 3d map registration using vision/laser and inertial sensing. In *ECMR*, 2007.
- [9] F.M. Mirzaei and S.I. Roumeliotis. A kalman filter-based algorithm for imu-camera calibration : Observability analysis and performance evaluation. volume 24, 2008.
- [10] J. A. Restrepo Specht. *Modélisation d'objets 3D par construction incrémentale d'un maillage triangulaire, dans un contexte Robotique*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, 2005.
- [11] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. of the Third International Conference on 3D Digital Imaging and Modeling*, 2001.
- [12] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.
- [13] J. Solà Ortega. *Towards Visual Localization, Mapping and Moving Objects Tracking by a Mobile Robot : a Geometric and Probabilistic Approach*. Thèse de doctorat, Institut National Polytechnique de Toulouse, Toulouse, France, 2007.
- [14] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. volume 13, 1994.