Vers la géolocalisation par vision d'une caméra mobile : exploitation d'un modèle 3D de ville et application au recalage visuel temps réel

Pierre Lothe, Steve Bourgeois, Fabien Dekeyser, Eric Royer, Michel Dhome

Vers la géolocalisation par vision d'une caméra mobile : exploitation d'un modèle 3D de ville et application au recalage visuel temps réel

Towards Geographical Referencing of Monocular SLAM Reconstruction Using 3D City Models : Application to Real-Time Accurate Vision-Based Localization

Pierre Lothe[1] Steve Bourgeois[1] Fabien Dekeyser[1] Eric Royer[2] Michel Dhome[2]

[1] CEA, LIST, Laboratoire Systèmes de Vision Embarqués
Boîte Courrier 94, Gif-sur-Yvette, F-91191 France ;
{pierre.lothe, steve.bourgeois, fabien.dekeyser}@cea.fr

[2] LASMEA, Université Blaise Pascal/CNRS
63177 Aubière, France
{eric.royer, michel.dhome}@lasmea.univ-bpclermont.fr

Résumé

Nous proposons un algorithme qui corrige a posteriori les dérives des méthodes de SLAM. Celui-ci exploite la connaissance a priori d'un modèle 3D simple de l'environnement. Notre méthode se déroule en deux étapes successives. Tout d'abord, un alignement grossier de la reconstruction SLAM avec le modèle 3D est calculé. Pour cela, nous proposons un algorithme d'ICP non-rigide exploitant un modèle de transformations articulées original et adapté au problème. L'alignement obtenu est ensuite raffiné à l'aide d'un ajustement de faisceaux. Pour cela, nous proposons une nouvelle fonction de coût permettant d'intégrer à la fois la cohérence entre les observations 2D et les points 3D reconstruits et la cohérence géométrique avec le modèle 3D de l'environnement. La méthode complète est validée sur des séquences de synthèse et réelles de grande échelle. Enfin, nous montrons que les reconstructions obtenues sont suffisamment précises pour être directement utilisées dans des applications de localisation globale.

Mots Clef

SLAM, reconstruction 3D, ICP non-rigide, ajustement de faisceaux, localisation globale.

Abstract

We propose a post processing algorithm that drastically reduces drift errors of SLAM methods. Our solution exploits a coarse 3D city model, for example from GIS database. First, we propose an original articulated transformation model in order to roughly align the SLAM reconstruction with this 3D model through a non-rigid ICP step. Then, to refine the reconstruction, we introduce a new bundle adjustment cost function that includes, in a single term, the usual 3D point/2D observation consistency constraint as well as the geometric constraints provided by the 3D model. Results on large-scale synthetic and real sequences show that our method successfully improves SLAM reconstructions. Besides, experiments prove that the resulting reconstruction is accurate enough to be directly used for global relocalization applications.

Keywords

SLAM, 3D reconstruction, non-rigid ICP, bundle adjustment, global localization.

1 Introduction

Les méthodes de localisation et cartographie simultanées (SLAM) par vision permettent de reconstruire à la fois l'environnement et la trajectoire d'une caméra mobile.
Différentes méthodes [5, 11], généralement basées sur l'usage du filtre de Kalman, ont été proposées pour effectuer cette tâche en temps réel. Néanmoins, en l'absence d'optimisation globale de la scène reconstruite, ces méthodes sont sensibles à l'accumulation d'erreurs. D'autre part, des méthodes réalisant une optimisation globale [12], généralement basées sur des ajustement de faisceaux, permettent de réduire cette dérive mais sans être temps-réel. Enfin, des méthodes d'optimisation sur une fenêtre temporelle glissante, basées sur un ajustement de faisceau incrémental [10] par exemple, permettent de réduire l'accumulation d'erreur et la mémoire utilisée tout en étant temps-réel.
Malgré ces progrès, les méthodes de SLAM par vision présentent encore des limitations : la trajectoire ainsi que le nuage de points reconstruit sont connus à une similitude près, ces données étant calculées dans un repère fixé arbitrairement. Il est donc impossible d'obtenir une localisation absolue des éléments reconstruits. D'autre part, en

plus d'être sensibles aux erreurs d'accumulation [5, 10], les méthodes de SLAM par vision monoculaire peuvent présenter une dérive du facteur d'échelle : les reconstructions sont réalisées à un facteur d'échelle près théoriquement constant sur la séquence entière, mais on observe souvent une dérive de ce facteur le long de la trajectoire. Même si ces erreurs peuvent être tolérées pour des applications de guidage utilisant des déplacements relatifs locaux, elles deviennent bloquantes pour les applications s'appuyant directement sur les reconstructions des méthodes de SLAM telles que la trajectométrie ou la localisation globale.
Par conséquent, nous proposons dans ce papier de corriger les reconstructions SLAM de grande échelle en utilisant un post-traitement qui introduit un modèle 3D simple de l'environnement.

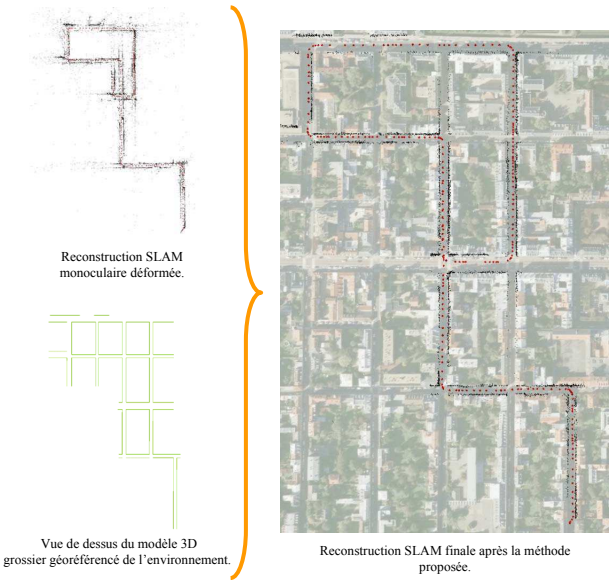


FIG. 1 – **Illustration de la méthode proposée.** Nous proposons de corriger la reconstruction SLAM initiale grâce à un modèle 3D de l'environnement. Le résultat obtenu est superposé à une image satellite.

2 Etat de l'art et contribution

Différentes approches ont été proposées pour éviter et corriger la dérive liée aux méthodes de SLAM.
Par exemple, Tardif *et al.* [15] montrent que l'utilisation de caméras omnidirectionnelles réduit considérablement la dérive de la trajectoire estimée. Cependant, cela augmente fortement le coût du matériel et la complexité de son intégration sur un véhicule grand public.
Pour corriger les reconstructions SLAM, une autre approche est d'utiliser une caméra bas-coût tout en introduisant des contraintes supplémentaires. Par exemple, Clemente *et al.* [4] utilisent la détection de boucles. Cependant, cette approche est uniquement efficace si la caméra passe de nombreuses fois au même endroit. De plus, aucune cohérence géométrique de la trajectoire n'est assurée : par exemple, une trajectoire qui décrit un carré peut en effet être reconstruite sous la forme de n'importe quel rectangle. Les contraintes supplémentaires peuvent également provenir d'un autre capteur ou d'une autre source d'informations. Levin *et al.* [8] proposent ainsi de corriger les reconstructions SLAM en introduisant un tracé grossier de la trajectoire. Une transformation par segments est appliquée à la reconstruction pour l'aligner au mieux avec ce tracé. Ensuite, un ajustement de faisceaux classique est réalisé pour corriger les erreurs locales résiduelles. Cependant, les informations apportées par le tracé grossier n'étant pas utilisées lors de l'ajustement de faisceaux, des problèmes de dérives peuvent apparaître à nouveau. Au contraire, Sourimant *et al.* [14] présentent une méthode de SLAM s'appuyant sur un modèle 3D grossier de l'environnement. En effet, ces ressources sont intéressantes puisqu'elles sont désormais largement distribuées (par exemple par les bases de données SIG) et qu'elles tendent à être normalisées[1]. La méthode proposée repose sur la reprojection des points d'intérêt de l'image courante sur ce modèle 3D pour reconstruire la scène et sur un algorithme de suivi KLT [16] pour calculer les déplacements de la caméra. Une limite de la méthode est que le modèle 3D n'est jamais remis en cause. La précision de la pose des caméras reconstruites est donc directement liée à la précision du modèle 3D.
A l'instar de Sourimant *et al.*, notre solution consiste à exploiter un modèle grossier de l'environnement (figure 1) afin de corriger des reconstructions SLAM de grande échelle. Cependant, nous considérons qu'exploiter un modèle 3D exempt d'erreur n'est pas une hypothèse réaliste. Nous introduisons donc dans cet article une méthode a posteriori en deux étapes qui fusionne une reconstruction SLAM avec le modèle 3D associé en prenant en compte les incertitudes de ces deux entrées. Notre première contribution est une méthode d'ICP non-rigide : nous utilisons une transformation de type articulée pour aligner au mieux le nuage de points reconstruits et le modèle 3D (section 3). Ensuite, une fonction de coût originale est proposée afin de corriger les erreurs résiduelles à travers un ajustement de faisceaux intégrant directement les informations liées au modèle 3D (section 4). Enfin, nous évaluons notre méthode à la fois sur des séquences de synthèse et réelles et proposons une application de localisation globale utilisant le nuage de points 3D ainsi corrigé (section 5).

3 Correction grossière des reconstructions SLAM

La méthode présentée dans cette partie est résumée dans la figure 2. Cette méthode prend en entrée la reconstruction initiale de SLAM [10], c'est à dire la trajectoire de la caméra et le nuage de points décrivant l'environnement (figure 2(b)). De plus, nous disposons d'un modèle 3D simple de l'environnement, simplement composé d'un ensemble

[1] http ://www.citygml.org/

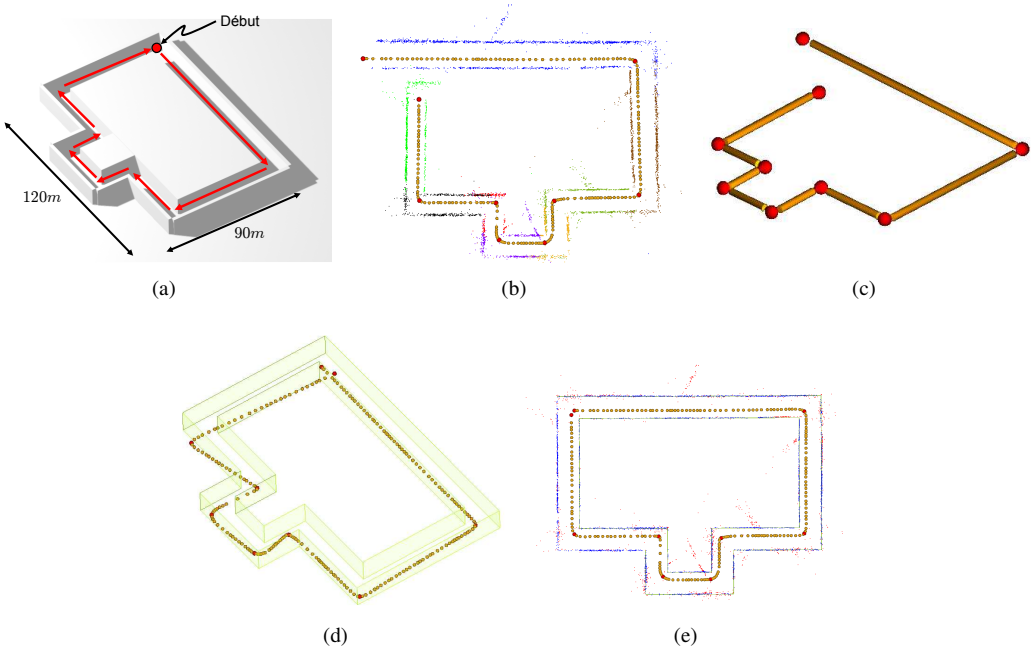


FIG. 2 – **Résumé de l'ICP non-rigide.** (a) Modèle CAO de synthèse et parcours suivi. (b) Reconstruction initiale en utilisant la méthode de SLAM de Mouragnon *et al.* [10]. La segmentation automatique de la reconstruction est également représentée : les sphères orange sont les caméras et les rouges correspondent aux extrémités des segments. Les points 3D reconstruits sont colorés selon leur appartenance à un fragment. (c) Le modèle de transformation proposé. La figure (d) représente l'initialisation de la trajectoire autour du modèle CAO (en vert) avant l'ICP. La figure (e) présente la reconstruction 3D après notre méthode : les points 3D bleus sont les points conservés par l'ICP et les rouges les points aberrants.

de plans verticaux représentant les différentes façades des bâtiments (figure 2(d)).

3.1 Méthodes de mise en correspondance

Pour corriger la géométrie de la reconstruction SLAM initiale, nous l'alignons avec le modèle 3D de l'environnement. Ce type de mise en correspondance a déjà été largement étudié et les méthodes les plus utilisées sont l'ICP (Iterative Closest Point) [13] et le Levenberg-Marquardt [6]. Cependant, ces méthodes ont été initialement créées pour des transformations (euclidiennes ou des similitudes) non-adaptées à notre problème, les déformations induites par les dérives étant beaucoup plus complexes. Des méthodes de mise en correspondance non-rigides sont également proposées de façon à corriger cette limite. En raison de leur grand nombre de degrés de liberté, ces algorithmes doivent être contraints (par exemple, par des termes de régularisation [3]).

Nous avons choisi de restreindre notre problème à une classe spécifique de transformations non-rigides approximant au mieux les déformations induites par le SLAM. Nous observons que le facteur d'échelle, pour le parcours classique d'un véhicule, est quasi-constant sur les lignes droites alors qu'il a tendance à changer dans les virages (voir la figure 2(b)). Nous avons donc décidé d'utiliser un modèle basé segments comme Levin [8] le propose : les lignes droites de la trajectoire sont considérées comme des éléments rigides et des articulations sont placées à chaque virage. Ainsi, les transformations retenues sont des similitudes par morceaux avec contraintes de jointure aux extrémités.

Tout d'abord, nous présentons la méthode de fragmentation de la reconstruction issue du SLAM. Ensuite, nous proposons une méthode ICP non-rigide permettant de trouver la transformation qui aligne au mieux le nuage de points reconstruit avec le modèle 3D.

3.2 Fragmentation de la reconstruction

L'étape de fragmentation consiste à segmenter la trajectoire de la caméra puis d'associer chacun des points 3D reconstruits à un des segments obtenus.

Pour cela, nous utilisons l'idée suggérée par Lowe [9] qui propose de découper récursivement un ensemble de points en différents segments en fonction à la fois de leur longueur et de leur déviation. Nous découpons donc la trajectoire reconstruite (représentée par un ensemble de caméras temporellement ordonnées) en m segments $(\mathcal{T}_i)_{1\leq i\leq m}$ dont les caméras extrémités sont notées $(\mathbf{e}_i, \mathbf{e}_{i+1})$.

Ensuite, pour associer chaque point 3D à un segment de trajectoire, nous définissons cette règle : un segment "voit" un point 3D si au moins une caméra de ce segment observe ce point. Deux cas de figure sont possibles. Le cas le plus simple apparait quand seul un segment voit le point : il est alors lié à ce segment. Si le point est observé par plusieurs segments, nous avons testé plusieurs politiques qui fournissent des résultats équivalents et nous avons donc choisi

arbitrairement d'associer le point au segment qui l'observe en dernier.
Nous appellerons désormais $\mathcal{B}_i$ un fragment composé des caméras de $\mathcal{T}_i$ (celles incluses entre $\mathbf{e}_i$ et $\mathbf{e}_{i+1}$) et des points 3D associés. Il est également à noter que pour $2 \leq i \leq$ m-1, $\mathcal{B}_i$ partage ses extrémités avec ses fragments voisins $\mathcal{B}_{i-1}$ et $\mathcal{B}_{i+1}$.

3.3 ICP non-rigide

Une fois les différents fragments obtenus, il est possible d'estimer la similitude par morceaux (avec contraintes aux extrémités) qui aligne au mieux le nuage de points 3D avec le modèle 3D. En pratique, ces transformations sont paramétrisées par les translations des extrémités $(\mathbf{e}_i)_{1\leq i\leq m+1}$. A partir de ces translations sont déduites les similitudes à appliquer à chacun des fragments (i.e. ses caméras et ses points 3D). La caméra étant embarquée sur un véhicule terrestre, nous choisissons de ne pas prendre en compte l'angle de roulis. Chaque extrémité $\mathbf{e}_i$ a 3 degrés de liberté et chaque segment en a donc bien 6.
Le problème que nous souhaitons résoudre est alors de trouver la position des extrémités qui permet de minimiser la distance entre les points 3D reconstruits $(\mathcal{Q}_i)_i$ et le modèle 3D $\mathcal{M}$, c'est à dire :

$$\min_{\mathbf{e}_1,...,\mathbf{e}_{m+1}} \sum_i d(\mathcal{Q}_i(\mathbf{e}_1,...,\mathbf{e}_{m+1}),\mathcal{M})^2 \qquad (1)$$

où d est la distance orthogonale entre $\mathcal{Q}_i(\mathbf{e}_1,...,\mathbf{e}_{m+1})$ (simplement noté $\mathcal{Q}_i$ dans la suite) et $\mathcal{M}$.

Association point-plan. La distance d devrait être la distance entre $\mathcal{Q}_i$ et le plan du modèle 3D auquel il appartient dans la réalité. Cependant, ce plan est inconnu dans notre cas. La distance d est donc calculée comme la distance entre $\mathcal{Q}_i$ et son plan le plus proche $\mathcal{P}_{h_i}$. Nous supposons que ce plan $\mathcal{P}_{h_i}$ ne change pas au cours de la minimisation :

$$\forall \mathcal{Q}_i, \mathcal{P}_{h_i} = \underset{\mathcal{P}\in\mathcal{M}}{argmin}\ d(\mathcal{Q}_i,\mathcal{P}) \qquad (2)$$

et le problème à résoudre devient donc :

$$\min_{\mathbf{e}_1,...,\mathbf{e}_{m+1}} \sum_i d(\mathcal{Q}_i,\mathcal{P}_{h_i})^2. \qquad (3)$$

Il est à noter que la distance d prend en compte que les plans 3D sont des plans finis : pour être associé au plan $\mathcal{P}$, un point 3D $\mathcal{Q}$ doit avoir sa projection orthogonale à l'intérieur des limites de $\mathcal{P}$.

Estimation robuste. L'association $(\mathcal{Q}_i,\mathcal{P}_{h_i})$ peut être erronée pour deux raisons : si $\mathcal{Q}_i$ est initialement trop éloigné de sa position réelle ou si ce point n'est pas sur un des plans du modèle CAO dans la réalité (i.e. n'appartient pas à une façade). Dans ces deux cas, le terme $d(\mathcal{Q}_i,\mathcal{P}_{h_i})$ peut empêcher l'obtention du minimum recherché. Pour limiter cet effet, nous utilisons un M-estimateur robuste ρ dans l'équation (3) :

$$\min_{\mathbf{e}_1,...,\mathbf{e}_m} \sum_i \rho(d(\mathcal{Q}_i,\mathcal{P}_{h_i})) \qquad (4)$$

Nous avons décidé d'utiliser le M-Estimateur de Tukey [2]. Le seuil du M-estimateur peut être réglé automatiquement grâce au MAD (Median Absolute Deviation). Le MAD fonctionne dans l'hypothèse où les données étudiées suivent une distribution gaussienne autour du modèle. Cette hypothèse peut être considérée correcte sur chacun des fragments pris séparément mais pas sur leur ensemble. Nous utilisons donc un seuil différent ξ_j par fragment. Nous avons donc à normaliser les valeurs du Tukey sur chaque segment :

$$\rho'_{l_i}(d(\mathcal{Q}_i,\mathcal{P}_{h_i})) = \frac{\rho_{l_i}(d(\mathcal{Q}_i,\mathcal{P}_{h_i}))}{\max\limits_{\mathcal{Q}_j\in\mathcal{B}_{l_i}} \rho_{l_i}(d(\mathcal{Q}_j,\mathcal{P}_{h_j}))} \qquad (5)$$

où l_i est l'indice du fragment contenant $\mathcal{Q}_i$ et ρ_{l_i} le M-estimateur de Tukey utilisé avec le seuil ξ_{l_i}.

Pondération des fragments. Avec la fonction de coût (5), chaque fragment a un poids dans la minimisation proportionnel au nombre de points 3D qu'il contient. Dans ce cas, les fragments possédant peu de points pourraient ne pas être optimisés en faveur des autres. Afin de donner le même poids à chacun des segments, nous avons décidé d'unifier les résidus de leurs points 3D en fonction de leur cardinal :

$$\rho^*_{l_i}(d(\mathcal{Q}_i,\mathcal{P}_{h_i})) = \frac{\rho'_{l_i}(d(\mathcal{Q}_i,\mathcal{P}_{h_i}))}{card(\mathcal{B}_{l_i})} \qquad (6)$$

et le problème s'écrit finalement :

$$\min_{\mathbf{e}_1,...,\mathbf{e}_{m+1}} \sum_i \rho^*_{l_i}(d(\mathcal{Q}_i,\mathcal{P}_{h_i})) \qquad (7)$$

problème que nous résolvons à l'aide de l'algorithme de Levenberg-Marquardt [7].

Optimisation itérative. En pratique, plusieurs minimisations non-linéaires sont réalisées, l'association point-plan étant recalculée avant chacune d'elles. Cela permet aux points 3D de changer le plan auquel ils sont associés tout en limitant les pertes de temps de calcul.

Initialisation. Les algorithmes de minimisation non-linéaires nécessitent une bonne initialisation des paramètres : la reconstruction 3D doit donc au moins être placée dans le même repère que le modèle CAO. Pour cela, estimer une transformation rigide est suffisant lorsque la dérive de la reconstruction 3D est faible. Cependant, la dérive du facteur d'échelle fréquemment observée dans les reconstructions de SLAM peut induire des déformations géométriques très importantes. Pour assurer la convergence de l'algorithme, nous avons donc décidé de placer grossièrement chaque extrémité $\mathbf{e}_i$ autour du modèle CAO. Cela peut être fait automatiquement si la séquence est synchronisée avec des données d'un système de navigation GPS grand public par exemple. Dans le cas contraire, cela peut être réalisé grâce à une interface graphique utilisateur.

4 Ajustement de faisceaux

Le modèle de transformation que nous utilisons dans l'étape précédente est trop contraint pour permettre une correction fine de la reconstruction SLAM : dans nos hypothèses, chaque segment de trajectoire est rigide et les caméras de ce segment ne peuvent donc pas se déplacer les unes par rapport aux autres. Des erreurs résiduelles de positionnement suivant la direction de la trajectoire peuvent être présentes sur chacun des segments. Les résultats expérimentaux mettent en évidence ce comportement (figure 4(a)).

Pour réduire ces erreurs résiduelles, il est nécessaire d'optimiser l'ensemble de la géométrie de la reconstruction. Ce type de problèmes est souvent résolu grâce à un ajustement de faisceaux. Cependant, en utilisant un ajustement de faisceaux classique qui ne minimise que les erreurs de reprojection dans les images, les contraintes existantes entre la reconstruction SLAM et le modèle 3D seraient perdues. Nous avons observé que cela peut conduire à des nouvelles déformations. C'est pourquoi nous proposons dans cette partie une fonction de coût originale qui inclut directement les informations liées au modèle 3D.

4.1 Fonction de coût proposée

Deux types d'informations doivent être introduites dans la fonction de coût : la relation entre les caméras et les points 3D qu'elles observent et la contrainte géométrique entre ces points et le modèle 3D. Une approche possible est d'ajouter ces deux types de résidus (donc de même unité de grandeur) à un facteur λ près. Cependant, ce facteur λ est généralement difficile à déterminer et spécifique à la séquence traitée. Dans la suite de cette partie, nous proposons donc une fonction de coût incluant les deux types d'informations dans un seul terme.

Cette fonction de coût est résumée dans la figure 3. L'idée principale est de forcer chacun des points 3D reconstruits à se situer sur un des plans du modèle. Tout d'abord, comme réalisé dans l'étape d'ICP non-rigide, chaque point $\mathcal{Q}_i$ est associé à son plan le plus proche $\mathcal{P}_{h_i}$. Considérons l'ensemble des caméras $(\mathcal{C}_i^j)_j$ qui observent $\mathcal{Q}_i$ et $(\mathbf{q}_i^j)_j$ ses observations 2D (figure 3(a)). Alors, pour chaque caméra $\mathcal{C}_i^j$, le point 3D $\mathcal{Q}_i^j$ est calculé comme étant l'intersection entre le faisceau issu de la rétroprojection de $\mathbf{q}_i^j$ et le plan $\mathcal{P}_{h_i}$ (figure 3(b)). Le barycentre des $(\mathcal{Q}_i^j)_j$ est noté $\mathcal{Q}'_i$ (figure 3(c)). $\mathcal{Q}'_i$ est donc le point 3D du plan $\mathcal{P}_{h_i}$ associé au point $\mathcal{Q}_i$. L'avantage d'utiliser $\mathcal{Q}'_i$, par exemple par rapport à projeter directement $\mathcal{Q}_i$ sur $\mathcal{P}_{h_i}$, est que le mouvement de $\mathcal{Q}'_i$ est directement lié aux déplacements des caméras l'observant. De plus, la position de $\mathcal{Q}_i$ n'étant pas utilisée dans la fonction de coût, nous n'avons pas à optimiser ses paramètres. La complexité du problème est donc fortement réduite. Ensuite, $\mathcal{Q}'_i$ est projeté dans chacune des caméras $\mathcal{C}_i^j$ et les points 2D obtenus sont notés $\mathbf{q'}_i^j$ (figure 3(d)). La fonction qui est minimisée est alors la somme au carré des distances entre les points $\mathbf{q}_i^j$ et les points $\mathbf{q'}_i^j$ en fonction des poses des caméras.

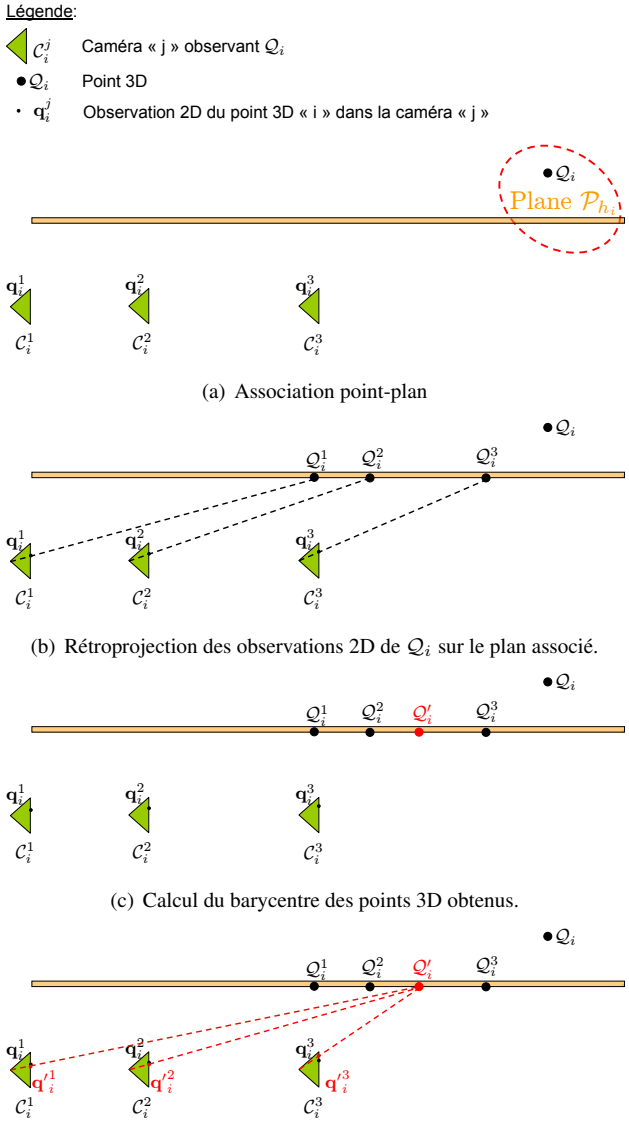


FIG. 3 – **Fonction de coût proposée.** Exemple d'un point 3D $\mathcal{Q}_i$ observé par 3 caméras. Les étapes successives sont décrites des sous-figures (a) à (d). Les résidus sont les distances 2D entre $(\mathbf{q}_i^j)_j$, les observations de $\mathcal{Q}_i$, et $(\mathbf{q'}_i^j)_j$, les projections de son point 3D associé $\mathcal{Q}'_i$.

4.2 Optimisation robuste

Pour être robuste aux points 3D aberrants, nous introduisons dans la fonction de coût à minimiser le M-estimateur de Geman-McClure [2] avec calcul automatique du seuil (basé sur le MAD).

La fonction de coût proposée ci-avant optimise la position des caméras pour les positions des $(\mathcal{Q}_i)_i$ estimées au préalable. Néanmoins, un mauvais positionnement des $(\mathcal{Q}_i)_i$ peut conduire à une mauvaise association point-plan et donc empêcher la minimisation (voir section 3.3). Afin d'outrepasser ce problème, il est nécessaire de mettre à jour la position des $(\mathcal{Q}_i)_i$ (qui n'est évidemment pas nécessai-

rement celle des $(\mathcal{Q}'_i)_i$) en fonction de la pose courante des caméras. Pour résoudre le problème global, il suffit alors d'itérer les étapes suivantes :

– Calcul des poses de caméras en minimisant la fonction de coût proposée dans la section 4.1 grâce à l'algorithme de Levenberg-Marquardt [7].
– Triangulation des points 3D $(\mathcal{Q}_i)_i$ à partir des nouvelles poses des caméras.
– Association de ces points 3D avec leur plan le plus proche.

La section suivante présente les résultats obtenus suite à l'ICP et l'ajustement de faisceaux.

5 Résultats expérimentaux

Dans cette section, nous présentons les résultats obtenus sur des séquences de synthèse et réelles. L'algorithme de SLAM utilisé est celui proposé par Mouragnon *et al.* [10]. Ensuite, nous présenterons les possibilités offertes par les reconstructions SLAM que nous avons corrigées à travers un exemple d'application de localisation globale.

5.1 Séquence de synthèse

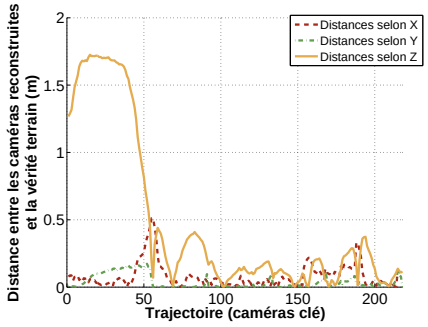
La figure 2 présente les différentes étapes de notre méthode. La séquence synthétique (basée pour le modèle 3D de la figure 2(a)) a été créée afin d'utiliser l'algorithme de SLAM dans un monde 3D texturé de synthèse. La trajectoire suivie est indiquée par les flèches rouges. De plus, le fait que la trajectoire de la caméra ne boucle pas dans la reconstruction initiale (figure 2(b)) souligne les problèmes de dérive inhérents aux méthodes de SLAM.

La première étape de notre méthode est l'ICP non-rigide. Pour son initialisation, nous avons simulé manuellement les résultats d'un système GPS bas coût : chacune des extrémités a été déplacée suivant l'erreur de ce type de système (figure 2(d)). Nous observons qu'après notre traitement (figure 2(e)) la boucle est restaurée alors qu'aucune contrainte de fermeture de boucles n'est directement insérée dans notre méthode. Cette amélioration est confirmée par les résultats numériques du tableau 5.1 : la distance moyenne entre les caméras reconstruites et la vérité terrain passe de plus de 4 mètres à environ 50 centimètres. Les statistiques avant l'ICP ont été calculées sur les 5591 points 3D reconstruits (parmi les 6848 proposés par le SLAM) considérés comme non-aberrants par l'étape d'ICP. De plus, il apparaît dans la figure 4(a) que seules les erreurs suivant la direction de la trajectoire restent signifiantes. Cela est directement lié au fait que les transformations non-rigides que nous utilisons supposent que le facteur d'échelle est strictement constant dans les lignes droites, ce qui est uniquement une approximation grossière.

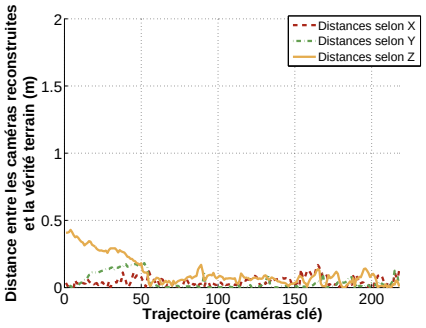
La figure 4(b) montre que l'étape originale d'ajustement de faisceaux permet de corriger ces erreurs résiduelles. L'erreur moyenne de positionnement des caméras atteint alors environ 14 centimètres, soit à peu près 3 fois moins qu'après l'ICP (table 5.1).

	Avant ICP	Après ICP	Après ajustement de faisceaux
Distance moyenne caméras-vérité terrain (m)	4.61	0.51	**0.14**
Ecart type (m)	2.25	0.59	**0.10**
Distance moyenne points 3D-modèle CAO (m)	3.37	0.11	**0.08**
Ecart type (m)	3.9	0.08	**0.08**
Seuil de Tukey	×	0.38	×

TAB. 1 – **Résultats numériques sur la séquence de synthèse.** Chaque valeur est une valeur moyenne sur l'ensemble de la reconstruction.



(a)

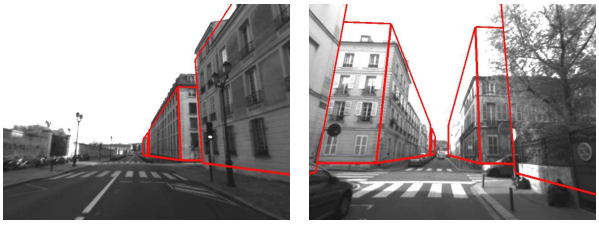


(b)

FIG. 4 – **Erreurs résiduelles sur les distances entre caméras reconstruites et vérité terrain.** Les coordonnées (X, Y, Z) sont relatives à chaque caméra : Z correspond à l'axe optique, X à la latitude et Y l'altitude. (a) représente ces résidus après l'étape d'ICP non-rigide et (b) après l'ajustement de faisceaux.

5.2 Séquence réelle

La séquence réelle est une vidéo (640x480) d'un parcours de 1500 mètres dans Versailles, France (voir figure 5(a)). La figure 5(b) indique que le modèle 3D est simplement composé d'un ensemble de plans verticaux représentant



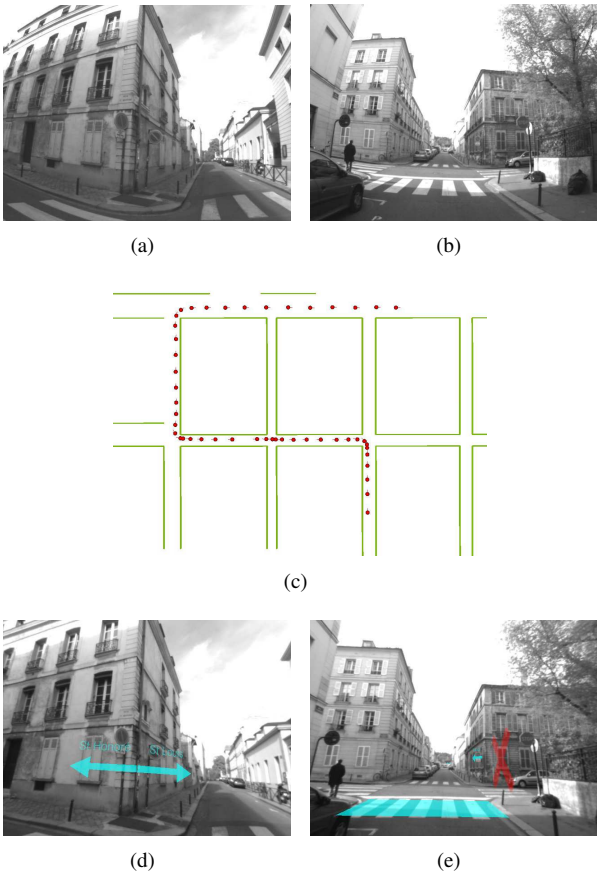
(a) (b)

FIG. 6 – **Projetion du modèle 3D dans les images clés.** Le modèle 3D utilisé peut être proche (a) ou éloigné (b) de la géométrie réelle de la scène.

grossièrement les façades des bâtiments. La précision de ce modèle 3D est environ de 2 mètres (figure 6(b)). La reconstruction SLAM initiale est un bon exemple de dérive : en la plaçant manuellement dans le même repère que le modèle 3D, il apparaît que la trajectoire devient incorrecte dès le troisième virage (figure 5(d)). Suite à notre méthode, la dérive est corrigée (voir figure 5(f)) sur l'ensemble de la trajectoire. En effet, la trajectoire de la caméra suit la route située entre les bâtiments et le nuage de points 3D retrouve sa cohérence spatiale. Ce résultat est confirmé par la superposition de la reconstruction SLAM l'image satellite correspondante (figure 1). De plus, la projection de modèle 3D dans chacune des images des caméras (figure 6) permet d'apprécier la précision des poses obtenues.

5.3 Application : localisation globale et réalité augmentée

Dans cette partie, nous proposons une application possible utilisant les résultats issus de notre méthode : la localisation globale par vision. En entrée, cette application utilise la reconstruction SLAM corrigée (et donc également géoréférencée) qui constituera la base de connaissances et un nouvel ensemble d'images prises dans les rues précédemment traitées. En sortie, la pose de la caméra pour chacune de ces images sera obtenue.
Il apparaît que la base de données proposée est bien adaptée pour une utilisation de localisation par vision. Tout d'abord, les points 3D étant reconstruits à partir d'une méthode de SLAM par vision, ils sont tous déjà associés à un descripteur de région d'intérêt. De plus, la base des points 3D est éparse et donc rapide à parcourir puisqu'uniquement composée des primitives pertinentes visuellement.
La figure 7 présente un exemple de localisation globale. Cette expérience est basée sur une nouvelle vidéo d'un véhicule traversant quatre des rues précédemment traitées. Les positions de ces images ont alors été calculées (indépendemment) à partir de notre base de données. Pour réaliser cette étape, pour chaque nouvelle image, nous cherchons tout d'abord l'image la plus proche dans la base de données. Alors, en utilisant la distance entre les descripteurs SURF [1], nous associons chacun des points d'intérêt de l'image courante est associé à un des points 3D visibles dans l'image clé retenue afin de calculer la pose de la caméra. Les résultats peuvent être appréciés dans la figure 7(c).
La précision des poses ainsi obtenues permet, par exemple, d'ajouter des données de réalité augmentée dans l'image courante (figures 7(d) et 7(e)), ces données supplémentaires ayant simplement à être ajoutées directement sur le modèle 3D.



(a) (b)

(c)

(d) (e)

FIG. 7 – **Réalité augmentée sur les images localisées.** La première ligne donne deux exemples d'images de la nouvelle vidéo. (b) est le résultat de la localisation globale de plusieurs images. Les images (c) et (d) présentent des résultats de réalité augmentée sur les images (a) et (b).

6 Conclusion

Dans ce papier, nous avons proposé une nouvelle approche pour corriger les reconstructions SLAM de grande échelle lorsqu'un modèle 3D grossier de l'environnement est disponible. Notre post-traitement repose sur deux étapes originales. Tout d'abord, nous utilisons une transformation par segments, obtenue grâce à une méthode ICP non-rigide, pour corriger grossièrement la reconstruction SLAM. Un ajustement de faisceaux spécifique, introduisant directement les informations liées au modèle 3D, est alors appliqué afin de raffiner la géométrie de la scène. Les expériences montrent que notre approche traite avec succès des séquences de synthèse et réelles. De plus, l'application

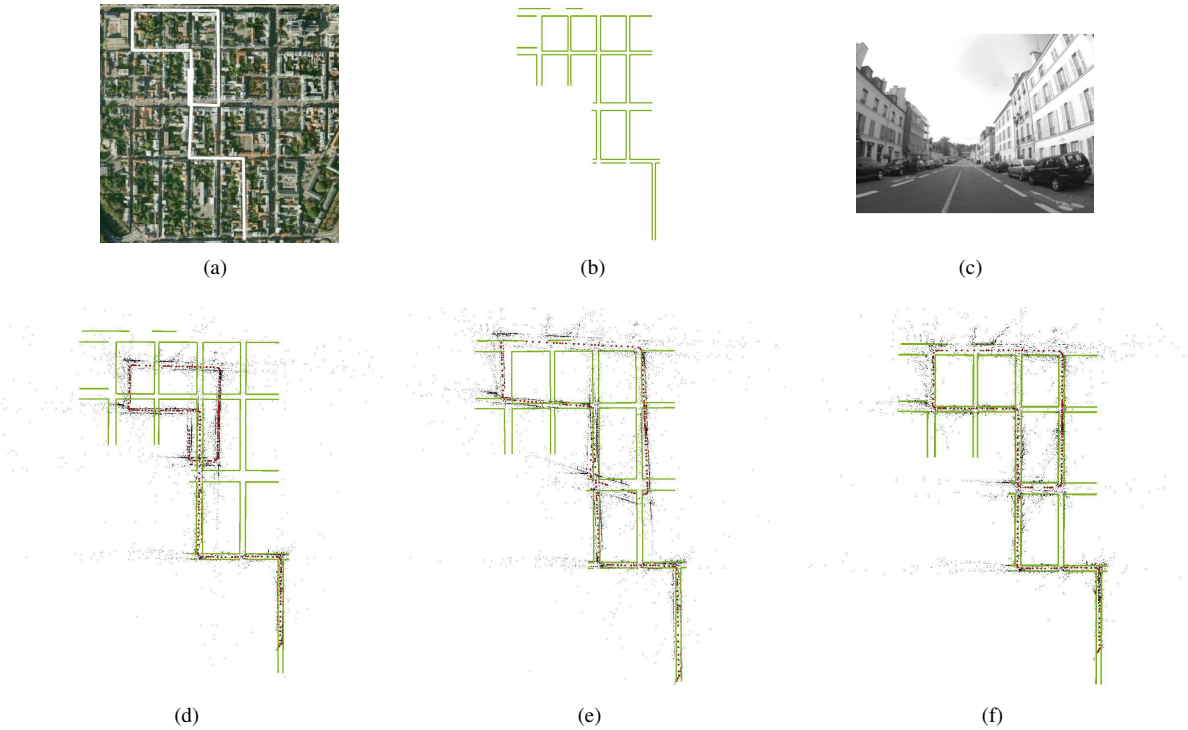


FIG. 5 – **Séquence de Versailles.** La première ligne présente les informations concernant la séquence réelle : la trajectoire suivie (a), le modèle 3D grossier (b) et une image issue de la vidéo traitée (c). La seconde ligne présente les différentes configurations de la reconstruction SLAM par rapport au modèle 3D : la reconstruction initiale avec l'algorithme de Mouragnon [10] (d), l'initialisation de la méthode ICP non-rigide (e) et le résultat de la méthode proposée (f).

de réalité augmentée proposée montre que la précision des reconstructions obtenues est suffisante pour qu'elles soient utilisées dans des problématiques de localisation globale.

Nos futurs travaux viseront à intégrer notre méthode directement dans le traitement en ligne du SLAM, afin de corriger progressivement la reconstruction et ainsi réaliser la localisation globale temps-réel sans avoir besoin de construire une base de données au préalable.

Références

[1] H. Bay, T. Tuytelaars, and L. V. Gool. Surf : Speeded up robust features. In *ECCV*, pages 346–359, 2006.

[2] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1) :57–91, 1996.

[3] U. Castellani, V. Gay-Bellile, and A. Bartoli. Joint reconstruction and registration of a deformable planar surface observed by a 3d sensor. In *3DIM*, pages 201–208, 2007.

[4] L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardos. Mapping Large Loops with a Single Hand-Held Camera. In *RSS*, 2007.

[5] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM : Real-time single camera SLAM. *PAMI*, 26(6) :1052–1067, 2007.

[6] A. Fitzgibbon. Robust registration of 2d and 3d point sets. In *BMVC*, pages 411–420, 2001.

[7] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, 2 :164–168, 1944.

[8] A. Levin and R. Szeliski. Visual odometry and map correlation. In *CVPR*, pages 611–618, 2004.

[9] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3) :355–395, 1987.

[10] E. Mouragnon, F. Dekeyser, P. Sayd, M. Lhuillier, and M. Dhome. Real time localization and 3d reconstruction. In *CVPR*, pages 363–370, 2006.

[11] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *CVPR*, pages 652–659, 2004.

[12] E. Royer, M. Lhuillier, M. Dhome, and T. Chateau. Localization in urban environments : Monocular vision compared to a differential gps sensor. In *CVPR*, pages 114–121, 2005.

[13] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3DIM*, pages 145–152, 2001.

[14] G. Sourimant, L. Morin, and K. Bouatouch. Gps, gis and video fusion for urban modeling. In *CGI*, may 2007.

[15] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *IROS*, pages 2531–2538, 2008.

[16] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon University, 1991.