Raffinement non linéaire d'une reconstruction de type SfM dans un environnement partiellement connu

Mohamed Tamaazousti[1] Vincent Gay-Bellile[1] Sylvie Naudet Collette[1] Michel Dhome[2]

[1]CEA, LIST, Laboratoire Vision et Ingénierie des Contenus
[2]LASMEA (CNRS / Université Blaise Pascal)

Point Courrier 94, Gif-sur-Yvette, F-91191 France
mohamed.tamaazousti@cea.fr

Résumé

Dans ce papier on s'intéresse à la localisation d'une caméra dans un environnement partiellement connu, c'est-à-dire pour lequel on dispose d'un modèle 3D géométrique d'une partie de la scène. Si la scène observée est statique, les éléments connue et inconnue de la scène fournissent des contraintes sur le mouvement de la caméra. Nous proposons un algorithme de raffinement non linéaire pour une reconstruction 3D obtenue par une méthode de type SfM. Cet algorithme prend en compte ces deux types d'informations dans une même fonction de coût permettant un raffinement plus précis et plus robuste. Ces affirmations sont démontrées sur différentes séquences de synthèse et réelles dans le cadre de deux applications : le suivi d'objet 3D et la localisation de véhicule en milieu urbain.

Mots Clef

Raffinement non linéaire, SfM, modèle 3D et ajustement de faisceaux.

Abstract

We address the challenging issue of camera localization in a partially known environment, i.e. for which a geometric 3D model that covers only a part of the observed scene is available. When this scene is static, both known and unknown parts of the environment provide constraints on the camera motion. This paper proposes a nonlinear refinement process of an initial SfM reconstruction that takes advantage of these two types of constraints. Compare to those that exploit only the model constraints i.e. the known part of the scene, including the unknown part of the environment in the optimization process yields a faster, more accurate and robust refinement. These statements are demonstrated on varied synthetic and real sequences for both 3D object tracking and outdoor localization applications.

Keywords

Nonlinear Refinement, SfM, 3D model and Bundle Adjustment.

1 Introduction

L'estimation de la pose d'une caméra par rapport à un objet d'intérêt est un sujet de recherche très actif. La plupart des méthodes existantes considèrent une caméra évoluant dans un environnement entièrement connu (un modèle 3D complet de la scène observée est disponible) ou entièrement inconnu. Les algorithmes de suivi basés modèle exploitent la connaissance à priori de la géométrie de l'objet pour estimer la pose de la caméra par rapport à ce dernier. Habituellement cette pose est estimée en temps réel en appariant des primitives extraites du modèle 3D de l'objet avec leur correspondant dans l'image courante [5]. Ce processus implique que l'objet d'intérêt soit observé au cours de toute la séquence ce qui ne convient pas aux grands environnements. D'autre part les méthodes de SfM (Structure from Motion) et de SLAM (Simultaneous Localization And Mapping) estiment le mouvement d'une caméra sans connaissance a priori de la géométrie de la scène observée. Ces algorithmes exploitent les relations multi-vues pour estimer le mouvement de la caméra, avec éventuellement une reconstruction 3D de la scène (sous forme de nuage de points 3D éparse). Des méthodes hors ligne [10, 12] puis en ligne [4, 8, 9] ont été proposées. Elles conviennent bien à de grands environnements. Malheureusement, en monoculaire ces algorithmes sont sujets à des accumulations d'erreurs et à une dérive du facteur d'échelle, ce qui réduit leurs domaines d'application. Récemment, des méthodes combinant une approche basée modèle et des techniques de SfM ont été proposées pour estimer avec précision le mouvement de la caméra dans un environnement partiellement connu. Bleser *et al.* dans [1] exploitent les contraintes géométrique du modèle pour initialiser (le repère et l'échelle) la reconstruction de l'algorithme SLAM. Le suivi est par la suite réalisé par une méthode "classique" de type SLAM qui ne tient plus compte du modèle 3D. La précision lors de l'initialisation n'est pas garantie, de plus la méthode reste sujette à des accumulations d'erreurs numériques et à une dérive du facteur d'échelle. Ces inconvénients sont particulièrement problématiques dans de grands environ-

nements. Pour résoudre ces problèmes, Lothe *et al.* introduisent dans [6] un processus en deux étapes. Dans un premier temps, une reconstruction standard de type SfM de tout l'environnement est approximativement alignée sur le modèle avec un ICP non rigide. Dans un second temps, un processus itératif combinant les relations multi-vues et les contraintes géométrique du modèle 3D est utilisé pour raffiner la reconstruction. Cependant le processus de raffinement ne prend pas en compte la partie inconnue de l'environnement. Ce qui limite sa précision et sa robustesse, notamment lorsque la caméra n'observe pas ou qu'en faible partie le modèle.

Dans ce papier, nous introduisons une méthode originale pour la localisation d'une caméra dans un environnement partiellement connu. Elle combine les informations géométriques de la partie connue et celle inconnue de l'environnement dans un algorithme de raffinement non linéaire. Il améliore la robustesse du raffinement et la précision de la reconstruction de tout l'environnement. Nous évaluons notre méthode sur des données synthétiques et réelles pour les deux types d'applications suivants : le suivi d'objet 3D et la localisation de véhicule en milieu urbain.

Plan. Dans la Section 2, nous introduisons les équations utilisées pour raffiner une reconstruction initiale pour un environnement connu et inconnu. Par la suite, nous proposons différentes solutions pour combiner de manière cohérente ces deux types de contraintes dans une seule fonction de coût. Elles sont présentées dans la Section 3 puis évaluées dans la Section 4. Dans la Section 5, cette méthode est appliquée pour des applications de suivi d'objet 3D et de localisation de véhicule.

Notation. Les matrices sont représentées par des caractères sans empattements comme M. Les vecteurs sont représentés en italique et exprimés en coordonnées homogènes, exemple $\mathbf{q} \sim (x, y, w)^\mathsf{T}$ où $^\mathsf{T}$ et la transposé et $\sim$ l'égalité à un facteur d'échelle près. Par la suite, nous supposons qu'une reconstruction initiale de la scène observée est préalablement estimée par un algorithme de type SfM comme par exemple [4, 8, 12]. Cette reconstruction est composée de N points 3D $\{\mathbf{Q}_i\}_{i=1}^N$ et de m caméras $\{\mathsf{C}_k\}_{k=1}^m$. Notons $\mathbf{q}_{i,k}$ l'observation du point 3D $\mathbf{Q}_i$ dans la caméra C_k et $\mathcal{A}_i$ l'ensemble des indices de caméras observant $\mathbf{Q}_i$. La matrice de projection P_k associée à la caméra C_k est donnée par $\mathsf{P}_k = \mathsf{K}\mathsf{R}_k^\mathsf{T}(\mathcal{I}_3| - \mathbf{t}_k)$, où K est la matrice des paramètres intrinsèques et $(\mathsf{R}_k, \mathbf{t}_k)$ les paramètres extrinsèques. Nous supposons qu'un modèle géométrique d'une partie de la scène observée est disponible. Il est composé d'un ensemble de plans π. La transformation entre le repère monde et le repère du modèle 3D est supposée approximativement connue (voir Section 5).

2 Raffinement non linéaire d'une reconstruction de type SfM

Le raffinement d'une reconstruction de type SfM repose sur la minimisation d'une fonction de coût non linéaire. Dans la suite nous décrivons les fonctions de coût utilisées dans le cas d'un environnement inconnu (Section 2.1) et connu (Section 2.2)

2.1 Raffinement non Linéaire dans un environnement inconnu

La technique standard pour raffiner une reconstruction de type SfM d'un environnement inconnu est l'ajustement de faisceaux (ou Bundle Adjustement BA) qui optimise simultanément les poses des caméras et la structure de la scène. Une alternative moins utilisée est de raffiner uniquement les poses des caméras à travers les contraintes de la géométrie épipolaire (EG) et ensuite de reconstruire un nuage de points 3D par triangulation.

Géométrie épipolaire (EG). Elle définit la relation entre les images acquises par deux caméras $(\mathsf{C}_1, \mathsf{C}_2)$ observant la même scène à partir de points de vues différents. La géométrie épipolaire [3] lie deux observations $(\mathbf{q}_{i,1}, \mathbf{q}_{i,2})$ d'un point 3D $\mathbf{Q}_i$ à travers la matrice Fondamentale : $\mathbf{q}_{i,2}^\mathsf{T}\mathsf{F}\mathbf{q}_{i,1} = 0$, où F est une matrice 3×3 de rang 2. Cette relation signifie qu'un point $\mathbf{q}_{i,2}$ dans la seconde image apparié au point $\mathbf{q}_{i,1}$ dans la première se trouve sur la ligne épipolaire $l = \mathsf{F}\mathbf{q}_{i,1}$. La matrice Fondamentale dépend directement du déplacement relatif entre deux poses de la caméra. On écrit alors la relation suivante : $\mathsf{F} = \mathsf{K}^{-\mathsf{T}}[\mathbf{t}_{1\to2}]_\times \mathsf{R}_{1\to2}\mathsf{K}^{-1}$. Ce déplacement inter caméras peut alors être raffinée en minimisant le critère non linéaire suivant : $\mathcal{E}((\mathsf{R}, \mathbf{t})_{1\to2}) = \sum_{i=1}^{N} d_l^2(\mathbf{q}_{i,2}, \mathsf{F}_{2,1}\mathbf{q}_{i,1}) + d_l^2(\mathbf{q}_{i,1}, \mathsf{F}_{1,2}\mathbf{q}_{i,2})$, où $d_l(\mathbf{q}, \mathbf{l})$ est la distance point-droite entre le point $\mathbf{q}$ et la droite $\mathbf{l}$, telle que $d_l^2(\mathbf{q}, \mathbf{l}) = \frac{(\mathbf{q}^\mathsf{T}\mathbf{l})^2}{\|\mathbf{l}\|^2 w^2}$. Ce principe peut être étendu au cas multi-vues. Le déplacement de la caméra est alors raffiné en minimisant la fonction de coût suivante :

$$\mathcal{E}\left(\left\{(\mathsf{R}, \mathbf{t})_{p\to p+1}\right\}_{p=1}^{m-1}\right) = \sum_{i=1}^{N}\sum_{j\in A_i}\sum_{k\in\mathcal{A}_i}^{k\neq j} d_l^2(\mathbf{q}_{i,j}, \mathsf{F}_{j,k}\mathbf{q}_{i,k}), \tag{1}$$

où, $\mathsf{F}_{j,k}$ est la matrice Fondamentale entre la paire d'images (j, k).

Ajustement de faisceaux (BA). L'ajustement de faisceaux [13] permet de raffiner simultanément les points 3D décrivant la scène observée et les poses de la caméra. Il minimise la somme des distances au carré entre les projections des points 3D dans les images et leurs observations. Cette distance géométrique est appelée l'erreur de reprojection. Les paramètres à optimiser sont les trois coordonnées des N points 3D et les six paramètres extrinsèques des m poses de la caméra. Le nombre total de paramètres est alors de $3N + 6m$. La fonction de coût du BA est donnée par :

$$\mathcal{E}\left(\{\mathsf{R}_k, \mathbf{t}_k\}_{k=1}^m, \{\mathbf{Q}_i\}_{i=1}^N\right) = \sum_{i=1}^{N}\sum_{j\in\mathcal{A}_i} d^2(\mathbf{q}_{i,j}, \mathsf{P}_j\mathbf{Q}_i), \tag{2}$$

où $d^2(\mathbf{q},\mathbf{q}') = \|\mathbf{q}-\mathbf{q}'\|^2$ est la distance point-point.

2.2 Raffinement non linéaire dans un environnement connu

Dans cette section, nous décrivons comment on peut exploiter des contraintes additionnelles fournies par un modèle 3D de l'environnement. Nous décrivons deux fonctions de coûts non linéaires qui combinent en un même terme, les relations multi vues et les contraintes d'appartenance au modèle 3D représenté par un ensemble de plans. Elles ont en commun de minimiser une erreur résiduelle exprimée en pixel.

Contraintes homographiques. Deux images observant un même plan π sont liées par une homographie H. Soit $\mathbf{q}_{i,1}$ l'observation d'un point $\mathbf{Q}_i \in \pi$ dans la première vue et $\mathbf{q}_{i,2}$ l'observation dans la seconde vue, alors $\mathbf{q}_{i,1} \sim \mathsf{H}\mathbf{q}_{i,2}$. Cette relation est l'équivalent de la géométrie épipolaire dans le cas planaire. L'homographie H induite par le plan π est donnée par :

$$\mathsf{H} = \mathsf{K}(\mathsf{R} - \frac{\mathbf{t}\mathbf{n}^\mathsf{T}}{d})\mathsf{K}^{-1}, \tag{3}$$

où, $\mathbf{n}$ représente la normale du plan et d la distance entre C_1 et le plan. Cette relation a été utilisée par Simon *et al.* dans [11] pour raffiner une reconstruction initiale de type SfM. La fonction de coût suivante est minimisée :

$$\mathcal{E}\left(\left\{(\mathsf{R},\mathbf{t})_{p\to p+1}\right\}_{p=1}^{m-1}\right) = \sum_{i=1}^{N}\sum_{j\in\mathcal{A}_i}\sum_{k\in\mathcal{A}_i}^{k\neq j} d^2(\mathbf{q}_{i,j}, \mathsf{H}_{j,k}^{\pi_i}\mathbf{q}_{i,k}), \tag{4}$$

où $\mathsf{H}_{j,k}^{\pi_i}$ est l'homographie induite par l'observation du plan π_i par les caméras j et k. Notons que cette fonction de coût ne prend pas en compte les points 3D. Seules les poses de caméras sont optimisées.

Ajustement de faisceaux avec contraintes au modèle. La fonction de coût décrite ci dessus inclut des contraintes au modèle à travers l'optimisation des déplacements inter caméra. Nous proposons une nouvelle fonction de coût qui contrairement à la première optimise également la structure 3D de la scène. L'idée principale est qu'un point 3D $\mathbf{Q}_i$ appartenant à un plan π_i a uniquement deux degrés de liberté. Soit M^{π_i} la matrice de transfert entre le repère du plan π_i et le repère du monde telle que $\mathbf{Q}_i = \mathsf{M}^{\pi_i}\mathbf{Q}_i^{\pi_i}$ où $\mathbf{Q}_i^{\pi_i} = (X^{\pi_i}, Y^{\pi_i}, 0, 1)^\mathsf{T}$ et (X^{π_i}, Y^{π_i}) sont les coordonnées de $\mathbf{Q}_i$ sur le repère du plan π_i. Cette relation peut être utilisée pour optimiser une reconstruction de type SfM avec des contraintes au modèle en minimisant la fonction de coût suivante :

$$\mathcal{E}\left(\{\mathsf{R}_j,\mathbf{t}_j\}_{j=1}^m, \{\mathbf{Q}_i^{\pi_i}\}_{i=1}^N\right) = \sum_{i=1}^{N}\sum_{j\in\mathcal{A}_i} d^2(\mathbf{q}_{i,j}, \mathsf{P}_j\mathsf{M}^{\pi_i}\mathbf{Q}_i^{\pi_i}). \tag{5}$$

En pratique, les points 3D reconstruits initialement par SfM n'appartiennent pas exactement aux plans du modèle. Une étape préliminaire est alors requise pour projeter chaque point 3D $\mathbf{Q}_i$ sur son plan π_i associé (voir la Section 3.3 pour plus de détails).

3 Raffinement non linéaire dans un environnement partiellement connu

Dans la section précédente, nous avons présenté différentes fonctions de coût pour raffiner une reconstruction initiale de type SfM d'un environnement connu (Eq. (4) et (5)) ou inconnu (Eq. (1), (2)). Dans cette section, nous décrivons comment fusionner dans un unique processus d'optimisation non linéaire les informations fournies par la partie connue et celle inconnue de l'environnement. Pour cela, une étape préliminaire de classification est requise pour décider quels points 3D $\mathbf{Q}_i$ de la reconstruction initiale appartiennent au modèle. L'association point-modèle est effectuée par lancé de rayon à partir des différentes observations $\{\mathbf{q}_{i,j}\}_{j\in A_i}$ de $\mathbf{Q}_i$. Card(A_i) votes sont alors obtenus pour les différents plans et le choix majoritaire est conservé. Une fois que la classification entre les parties de l'environnement est effectuée, les points 3D associés à un plan π_i du modèle sont projetés sur ce dernier avant la minimisation de l'Eq. 5 (pas nécessaire pour Eq. 4 qui n'optimise pas les points 3D). Le barycentre des intersections entre les lancés de rayons et le plan π_i est sélectionné comme étant la position initiale du point 3D. Notons $\mathcal{M}$ l'ensemble des indices des points 3D associés au modèle et $\mathcal{U}$ l'ensemble des indices des autres points 3D qui constituent la partie inconnue de l'environnement, avec card$(\mathcal{M})$ + card$(\mathcal{U}) = N$.

3.1 Estimation robuste

Un mauvais recalage initial et la dérive induite par les algorithmes de type SfM peuvent introduire de mauvaises associations point-modèle qui peuvent empêcher la convergence du processus d'optimisation. Pour gérer ces associations aberrantes une estimation robuste est effectuée avec le M-estimateur de Geman-McClure $\rho(r,c) : \mathbb{R} \to [0 \cdot\cdot 1]$ où

$$\rho(r,c) = \frac{r^2}{r^2+c^2}, \tag{8}$$

avec, r est l'erreur résiduelle d'une des fonctions de coût décrites précédemment (1), (2), (4) ou (5), et c le seuil de rejet. Il est estimé automatiquement en utilisant la médiane des valeurs absolue (Median Absolute Deviation MAD) tel que $c = median(\mathbf{r}) + 1.4826\text{MAD}(\mathbf{r})$ où $\mathbf{r}$ est un vecteur concaténant les résidus d'une des fonctions de coût. Notons que le MAD suppose une distribution normale des résidus.

3.2 Fonctions de coût composées

Combiner Eq. (1) ou (2) avec Eq. (4) ou (5) n'est pas évident même si les différents résidus sont tous exprimés dans la même unité (pixel). En effet, elles ne sont pas nécessairement du même ordre de grandeur : les erreurs résiduelles associées à la partie connue de l'environnement sont généralement plus élevées. La combinaison de la partie connue avec celle inconnue de l'environnement est modélisée ici comme un problème de moindre carré bi-objectif.

$$\mathcal{E}\left(\left\{(\mathsf{R},\mathbf{t})_{p\to p+1}\right\}_{p=1}^{m-1}\right)=\underbrace{\sum_{i\in\mathcal{U}}\sum_{j\in A_i}\sum_{k\in A_i}^{k\neq j}\rho\left(d_l^2(\mathbf{q}_{i,j},\mathsf{F}_{j,k}\mathbf{q}_{i,k}),c_1\right)}_{\text{Partie inconnue de l'environnement (E)}}+\underbrace{\sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{A}_i}\sum_{k\in\mathcal{A}_i}^{k\neq j}\rho\left(d^2(\mathbf{q}_{i,j},\mathsf{H}_{j,k}^{\pi_i}\mathbf{q}_{i,k}),c_2\right)}_{\text{Partie connue de l'environnement (M)}} \tag{6}$$

$$\mathcal{E}\left(\{\mathsf{R}_j,\mathbf{t}_j\}_{j=1}^{m},\{\mathbf{Q}_i\}_{i\in\mathcal{U}},\{\mathbf{Q}_i^{\pi_i}\}_{i\in\mathcal{M}}\right)=\underbrace{\sum_{i\in\mathcal{U}}\sum_{j\in\mathcal{A}_i}\rho\left(d^2(\mathbf{q}_{i,j},\mathsf{P}_j\mathbf{Q}_i),c_1\right)}_{\text{Partie inconnue de l'environnement (E)}}+\underbrace{\sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{A}_i}\rho\left(d^2(\mathbf{q}_{i,j},\mathsf{P}_j\mathsf{M}^{\pi_i}\mathbf{Q}_i^{\pi_i}),c_2\right)}_{\text{Partie connue de l'environnement (M)}} \tag{7}$$

Cohérence dans les combinaisons. Par souci de cohérence du choix de combinaisons nous ne présentons ici que deux fonctions de coût composées. La première, utilise les contraintes homographiques (Eq. (4)) pour la partie de l'environnement associé au modèle et son équivalent dans le cas d'une structure inconnue c'est-à-dire la géométrie épipolaire (Eq. (1)). Elles ont en commun le fait de ne minimiser que les déplacements relatifs entre deux caméras. La seconde fonction de coût est composée des Eq. (2) et (5) qui utilisent explicitement les points 3D dans un ajustement de faisceaux. Ainsi, les points 3D associés au modèle 3D ont uniquement deux degrés de liberté alors que les points 3D de la partie inconnue de l'environnement ont trois degrés de liberté.

Pondération avec l'estimateur robuste. L'une des grandes difficultés dans la minimisation d'un problème bi-objectif est de contrôler l'influence de chaque terme. Cela est généralement effectué en utilisant un paramètre de pondération qui est fixé expérimentalement ou alors en utilisant la validation croisée [2]. Nous proposons ici une alternative plus simple : l'influence de chaque terme est directement contrôlée en utilisant le seuil de rejet de l'estimateur robuste. Nous avons vu précédemment qu'un estimateur robuste est utilisé pour gérer les mauvaises associations point-modèle, pour les fonctions de coût de la partie connue de l'environnement. Nous proposons alors les deux fonctions de coût données par Eq. (6) et (7). Notons que comme l'estimateur de Geman-McClure normalise les résidus, nous appliquons également l'estimateur robuste aux fonctions de coût de la partie inconnue de l'environnement.

Il existe alors plusieurs possibilités pour contrôler l'influence de chaque terme avec le seuil de rejet. Nous en avons exploré trois :

– combinaison 1 : $c_1 = c_{Env}$ et $c_2 = c_{Modele}$
– combinaison 2 : $c_1 = c_2 = c_{All}$
– combinaison 3 : $c_1 = c_2 = c_{Modele}$

où, c_{Modele} est le seuil de rejet estimé sur les résidus du modèle comme ceux utilisés dans Eq. (4), (5), c_{Env} est le seuil estimé sur les résidus de la partie inconnue de l'environnement tels que ceux utilisés dans Eq. (1) ou (2) et c_{All} est estimé sur l'ensemble des résidus. Pour la combinaison 1 il y a un seuil de rejet associé à chaque terme alors que pour les combinaisons 2 et 3 un seul seuil est déterminé. La différence est que pour la combinaison 2 il est évalué sur l'ensemble des résidus alors que pour la combinaison 3 seuls les résidus associés à la partie connue de l'environnement ont contribué à son estimation. La combinaison 2 considère que ces deux types de résidus ont le même ordre de grandeur. Les combinaisons 1 et 3 font au contraire l'hypothèse que les résidus associés à la partie connue de l'environnement ont généralement des valeurs plus élevées. La combinaison 1 traite les deux parties de l'environnement de manière identique alors que la combinaison 3 favorise la partie modélisée au cours du processus d'optimisation. Dans ce dernier cas la plupart des points de la partie inconnue sont conservés du fait de leur valeur plus faible ce qui garantie que les contraintes de l'environnement restent vérifiées. Ces trois combinaisons sont évaluées sur des données de synthèse dans la Section 4.2.

3.3 Optimisation itérative

Au cours du processus d'optimisation les associations point-modèle et le seuil de rejet de l'estimateur robuste doivent être réestimés pour garantir une convergence optimale. Les étapes suivantes sont alors itérées jusqu'à convergence :

1. Association des points 3D $(\mathbf{Q}_i)_{i\in\mathcal{U}\cup\mathcal{M}}$ à la partie connue ou inconnue de l'environnement.
2. Projection des points 3D $(\mathbf{Q}_i)_{i\in\mathcal{M}}$ sur le plan auquel ils sont associés π_i (Cette étape est réalisée uniquement pour Eq. (7)).
3. Calcul des seuils de rejet c_1 et c_2.
4. Minimisation de (6) ou (7) par l'algorithme de Levenberg Marquardt (LM) [7] (quelques itérations).
5. Triangulation des points 3D $(\mathbf{Q}_i)_{i\in\mathcal{U}\cup\mathcal{M}}$ pour Eq. (6) et $(\mathbf{Q}_i)_{i\in\mathcal{M}}$ [1] pour Eq. (7) avec les poses des caméras estimées.

4 Évaluation sur des données de synthèse

Dans cette section nous comparons quatre algorithmes sur une séquence de synthèse générée avec un logiciel de modélisation 3D : BA_M, BA_M&E, EG_M and EG_M&E [2]. Elles minimisent respectivement les Eq. (5), (7), (4) et (6) en suivant la procédure décrite en Section 3.3. Notons que

1. Pour Eq. (7), les points 3D de la partie connue de l'environnement doivent être triangulés pour la réestimation des associations point-modèle.

2. M signifie que seules les contraintes au modèle (c'est-à-dire de la partie connue de l'environnement) sont utilisées tandis que M&E signifie qu'à la fois les contraintes au modèle mais aussi les informations du reste de l'environnement sont prises en compte.

pour BA_M et EG_M le M-estimateur de Geman-Mclure (Section 3.1) est aussi utilisé pour gérer les mauvaises associations point-modèle. Dans un premier temps une comparaison des trois choix de combinaison de la Section 3.2 est effectuée avec l'algorithme BA_M&E. Dans un second temps, une comparaison poussée des quatre algorithmes décrits précédemment est réalisée. La comparaison est réalisée en terme de bassin de convergence, de vitesse de convergence, et aussi sur la précision de la reconstruction obtenue ainsi que sur la robustesse à un mauvais recalage initial.

4.1 La séquence du cube

Cette séquence, représentée sur la Figure 1 est composée d'un cube principal partiellement occulté par d'autres petits cubes et situé sur un sol texturé. L'objet d'intérêt, c'est-à-dire pour lequel un modèle 3D est disponible, est le cube principal. L'environnement inconnu est alors composé du sol et des petits cubes situés autour. La trajectoire de la caméra est un cercle de 3 mètres de rayon autour du cube principal.

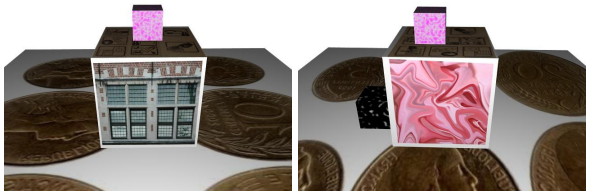

FIGURE 1 – Illustration de la séquence du cube.

4.2 Choix de la combinaison

Pour l'algorithme de BA_M&E nous comparons les trois propositions de combinaisons décrites en Section 3.2. La reconstruction initiale est obtenue avec l'algorithme de SfM décrit dans [8]. Le repère du monde et l'échelle de la reconstruction sont fixés avec la vérité terrain. Cette reconstruction initiale est alors raffinée en minimisant l'Eq. (7) avec l'une des trois combinaisons en suivant la procédure décrite en Section 3.3.

La Figure 2 (Gauche) illustre la distribution des erreurs pour les deux types de résidus après l'étape 2 de la procédure de minimisation, voir la Section 3.3. L'ordre de grandeur des erreurs résiduelles associées à la partie connue de l'environnement est plus élevé que celle de la partie inconnue, comme pressenti. Cela explique que la combinaison 2 présente le plus mauvais résultat comme le montre la Figure 2 (Droite). Comme son seuil de rejet est sous estimé : la plupart des résidus de la partie modélisée sont rejetés par l'estimateur robuste. La minimisation de l'Eq.(5) (qui correspond à l'algorithme BA_M) avec la combinaison 1 donne des résultats similaires. Finalement la combinaison 3 donne les meilleurs résultats car elle permet de conserver la plupart des points de la partie inconnue de l'environnement. Cela s'explique par le fait que les contraintes de la composante non modélisée de l'environnement restent vérifiées.

Notons que des résultats similaires ont été obtenus sur d'autres séquences de synthèse et également avec l'algorithme EG_M&E. Ils ne sont pas présentés ici pour éviter de surcharger le papier. Dans la suite nous ne considérerons plus que la combinaison 3 pour les Eq.(6) et (7).

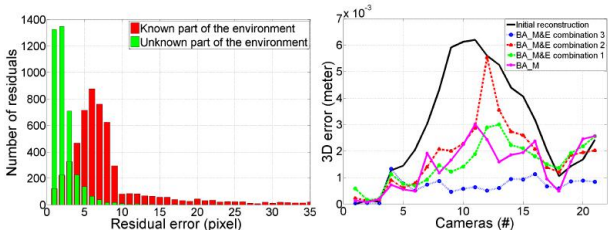

FIGURE 2 – A gauche : distributions des erreurs résiduelles. En vert, (resp. en rouge) la distribution des erreurs résiduelles associées à la composante inconnue (resp. connue). A droite : Erreurs sur les positions de la caméra exprimées en mètre pour les différentes combinaisons.

4.3 Comparaison des quatre algorithmes.

Protocole expérimental. Nous comparons ici les quatre algorithmes (EG_M, BA_M, EG_M&E et BA_M&E) en simulant différentes sources d'erreurs comme celle du recalage initial, de la dérive, *etc*. A partir des poses de caméras de la vérité terrain, un nuage de points 3D éparse est généré par triangulation de points d'intérêt appariés au cours de la séquence. Deux types de perturbations sont alors générés sur cette reconstruction initiale.

- Le premier test (TEST RIGIDE) simule des imprécisions du recalage initial entre les repères du monde et du modèle. Pour cela, une perturbation rigide est appliquée sur la reconstruction globale (caméras et points 3D).
- Le deuxième test (TEST NON RIGIDE) simule les erreurs d'estimation du déplacement relatif entre deux caméras. Elles ont pour origine les algorithmes de type SfM en présence de bruit, de valeurs aberrantes, et les dérives numérique, *etc*. Une perturbation non rigide de la reconstruction globale est réalisée en perturbant de manière aléatoire les déplacements inter caméras et en régénérant par la suite un nuage de points 3D par triangulation.

L'amplitude de perturbation varie de 1% à 10% du rayon du cercle décrivant la trajectoire de la caméra. On applique alors les quatre algorithmes sur les reconstructions ainsi obtenues. La qualité de la reconstruction finale est mesurée par le RMS 3D sur les positions de la caméra entre celles de la vérité terrain et celles estimées. Les quatre algorithmes de raffinement sont donc comparés en termes de précision, vitesse et fréquence de convergence. La précision est donnée par la valeur du RMS 3D uniquement lorsqu'il y a convergence de l'algorithme. La fréquence de convergence est le pourcentage de tirages pour lesquels le

RMS 3D a diminué. La vitesse de convergence est mesurée par l'évolution de l'erreur 3D au cours des itérations successives du LM, pour une perturbation donnée (4% dans nos expériences). Les résultats sont illustrés sur la Figure 3. Une moyenne a été effectuée sur 50 tirages aléatoires.

Fréquence de convergence. Les quatre algorithmes ont un comportement similaire pour TEST RIGIDE et TEST NON RIGIDE. EG_M&E et BA_M&E ont le bassin de convergence le plus large alors que BA_M a le plus petit bassin de convergence. Pour un déplacement d'amplitude 10% lors du TEST NON RIGID, EG_M&E et BA_M&E convergent approximativement dans tous les cas, alors que EG_M converge a 60% et BA_M ne converge jamais.

Précision. EG_M&E et BA_M&E sont les algorithmes les plus précis pour des perturbations rigides et non rigides. Ils sont suivis de près par EG_M alors que BA_M a les plus mauvaises performances. Par exemple, pour une amplitude de perturbation de 8% pour le test rigide, le RMS 3D de EG_M&E et BA_M&E sont inférieur à 5cm. Il est supérieur à 5cm pour EG_M et autour de 11cm pour BA_M.

Vitesse de convergence. BA_M&E converge plus vite que les autres algorithmes. Pour TEST RIGIDE, après 3 itérations BA_M&E réduit l'erreur 3D par un facteur 2.7, EG_M&E par un facteur 2 et seulement par un facteur 1.5 pour EG_M et BA_M.

Résumé. BA_M&E et EG_M&E sont plus performantes que BA_M et EG_M en termes de précision, vitesse et fréquence de convergence. Cela montre que la prise en compte de la partie inconnue de l'environnement dans le raffinement non linéaire améliore de manière remarquable les résultats.

5 Application à la localisation dans de grands environnements et au suivi d'objets 3D

Dans cette section nous évaluons le processus sur des données réelles pour deux applications : la localisation dans un grand environnement et le suivi d'objet 3D. Deux séquences réelles ont été réalisées avec une caméra IEEE1394 GUPPY fournissant des images (640×480) à 30 images par seconde.

5.1 Localisation en milieu urbain

Des travaux récents, comme par exemple [6], ont montré que dans un contexte urbain on peut utiliser un modèle de type SIG pour contraindre le nuage de points provenant d'un algorithme de type SfM pour obtenir une reconstruction plus précise. Dans ce papier, ils proposent un processus hors ligne en deux étapes qui aligne dans un premier temps approximativement la reconstruction sur le modèle avec un ICP non rigide et par la suite raffine la reconstruction avec une optimisation non linéaire basée modèle. Notons que la partie inconnue de l'environnement c'est-à-dire tout ce qui n'est pas un bâtiment n'est pas pris en compte

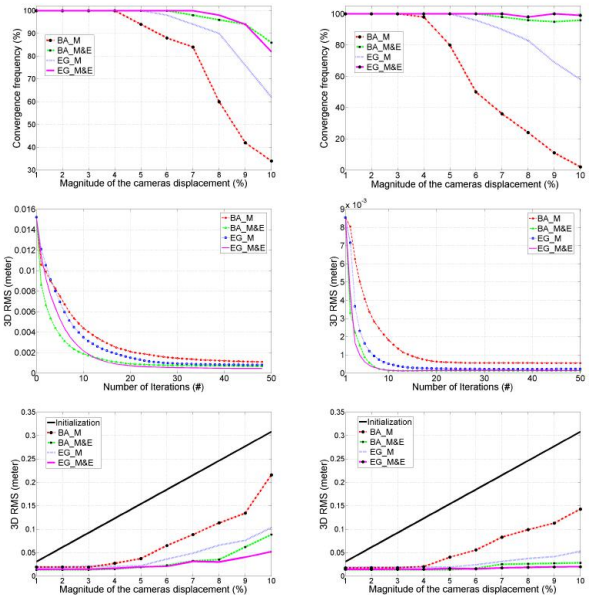

FIGURE 3 – Résultats obtenus avec les quatre algorithmes BA_M, BA_M&E, EG_M et EG_M&E pour TEST RIGIDE (Gauche) et TEST NON RIGIDE (Droite).

dans leur processus d'optimisation. Leur algorithme de raffinement non linéaire est similaire à BA_M et EG_M avec une fonction de coût légèrement différente.
Ce type d'algorithme de raffinement n'est évidemment pas robuste lorsque la partie connue de l'environnement est occultée ou absente. Cependant, ce cas de figure est très commun en milieu urbain : il n'y a pas des bâtiments de chaque côté de la chaussée dans toutes les rues et ils peuvent être occultés par des bus, des camions *etc*. Nous montrons par la suite que le fait d'intégrer, dans le processus de minimisation, la partie inconnue de l'environnement c'est-à-dire la route, les arbres, *etc*, permet de résoudre ces problèmes.
Une séquence vidéo de 975 images a été acquise au cours d'un déplacement en voiture de 500 mètres. La reconstruction initiale de type SfM a été approximativement alignée sur le modèle avec une correction par ICP non rigide comme dans [6]. Elle est alors raffinée par les algorithmes EG_M&E et EG_M. Les reconstructions obtenues sont alors évaluées qualitativement par une relocalisation en ligne.

Raffinement non linéaire hors ligne. La Figure 4 illustre le nuage de points 3D et la trajectoire de la caméra obtenus après le raffinement par les algorithmes EG_M et EG_M&E. Les principales différences entre les deux sont localisées à deux endroits de la scène et ont été entourées et zoomées. Dans le premier cas, la caméra au début du carrefour n'observe pas de bâtiment et pour le deuxième, les bâtiments sur le coté gauche sont occultés par un bus. Dans ces deux cas critiques avec l'algorithme EG_M la caméra présente une trajectoire improbable et la structure du nuage de points est aussi erronée (en rouge sur la Figure 4).

L'algorithme EG_M&E fournit une trajectoire régulière, en particulier dans la première zone où la structure du mur semble être rétablie (en bleu sur la Figure 4). Ces résultats montrent qu'en utilisant la partie inconnue de l'environnement, la reconstruction est dans l'ensemble de meilleur qualité [3]. Ces deux reconstructions sont alors utilisées pour construire deux bases de données qui encodent les descripteurs de l'ensemble des points 3D [4] sous la forme d'un arbre de vocabulaire.

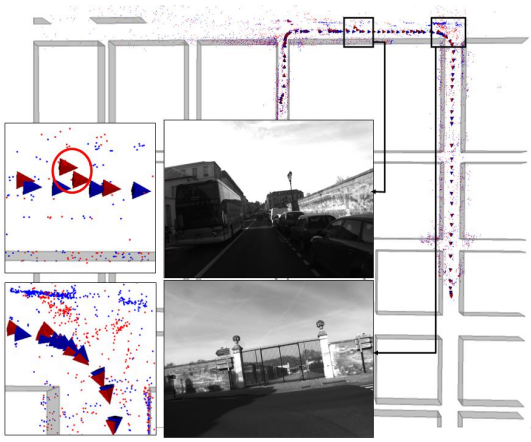

FIGURE 4 – Reconstruction par SfM en milieu urbain. En bleu (resp. rouge) la reconstruction obtenue après raffinement avec l'algorithme EG_M&E (resp. EG_M). Les cercles rouges mettent en avant les poses improbables de la caméra introduites par l'algorithme EG_M.

Relocalisation en ligne. Une autre séquence vidéo de 649 images a été réalisée en suivant approximativement la même trajectoire. L'environnement a légèrement changé depuis la précédente séquence, par exemple des voitures garées sur la chaussée. Pour chaque image de la séquence une relocalisation est effectuée. Pour cela une mise en correspondance entre les descripteurs extraits de l'image courante et ceux de la base de donnée est effectuée. La pose de la caméra est alors calculée avec l'estimateur RANSAC sur les appariements. Le nombre de relocalisation retenues est de 539 et 642 pour les bases de données obtenues respectivement avec les algorithmes EG_M et EG_M&E. Avec la première base de donnée, des échecs de relocalisation apparaissent principalement dans les deux parties de la séquence décrite précédemment. De plus, le nombre de correspondances 2D/3D utilisées pour calculer la pose de la caméra est de 50 pour EG_M contre 60 pour EG_M&E en moyenne sur la séquence. Nous pouvons alors conclure que l'ensemble des points 3D est plus cohérent pour la reconstruction raffinée par l'algorithme EG_M&E. Cela montre que l'algorithme EG_M&E fournit une reconstruction de meilleure qualité.

3. Des conclusions similaires ont été obtenues avec les algorithmes BA_M&E et BA_M.

4. Les points 3D associés à la partie connue et inconnue de l'environnement sont conservés pour les deux cartes.

5.2 Suivi d'objet 3D

L'objet d'intérêt est un modèle réduit de la Citroën C4 de Sebastien Loeb du championnat WRC. Le modèle 3D utilisé dans nos expériences est composé de 1600 triangles. Il n'inclus pas certaines pièces de la voiture comme les roues, l'aileron, les fenêtres, *etc*. Nous avons placé la voiture sur un bureau composé d'un écran et clavier d'ordinateur, de livres, *etc*. Ils constituent la partie inconnue de l'environnement.

Comparaison de trois algorithmes pour le suivi. Nous comparons alors les trois algorithmes de raffinement non linéaires : l'ajustement de faisceaux (BA) avec BA_M et BA_M&E sur une séquence difficile. Elle présente de grandes variations en échelle, des variations de luminosité, des occultations partielles et totales de la voiture, *etc*. Le recalage initial est réalisé en appariant la première image de la séquence avec une image clef qui a été recalée au préalable sur le modèle.

Résultats. La Figure 5 présente les résultats obtenus par les trois algorithmes de raffinement utilisés sur cette séquence. Le recalage initial semble précis sur la première image (le modèle se projette précisément) mais en tournant autour de la voiture nous observons que ce n'est pas réellement le cas. L'algorithme BA de raffinement ne peut pas corriger cette imprécision comme on le voit sur la Figure 5 (en haut à gauche) et conserve donc cette erreur au cours de toute la séquence. Les algorithmes de raffinement BA_M&E et BA_M réussissent à corriger cette erreur de recalage après quelques images : l'avant et l'arrière de la voiture se projettent parfaitement dans les images. Notons que l'algorithme de raffinement BA_M&E donne de meilleurs résultats que BA_M quand l'objet d'intérêt est occulté ou qu'il ne prend qu'une petite partie des images comme le montre la Figure 5 (à droite). Il gère avec succès et précision, la localisation de la caméra au cours de toute la séquence. La combinaison des informations fournies par la partie connue et inconnue de l'environnement permet donc une localisation plus précise et robuste.

6 Conclusion

Nous avons présenté différents algorithmes de raffinement non linéaires d'une reconstruction de type SfM pour un environnement partiellement connu. Deux fonctions de coût prenant en compte l'information fournie par la partie connue et celle inconnue de l'environnement, ont été proposées. Leur combinaison optimale ainsi que le processus d'optimisation ont été attentivement étudiés. Des tests sur données de synthèse et réelles montrent l'apport de la partie inconnue de l'environnement dans le raffinement. Cela permet d'augmenter la précision de la reconstruction, la robustesse et le bassin de convergence. Nous avons appliqué avec succès ce processus au suivi d'objet 3D et à la localisation en milieu urbain. Pour les perspectives nous souhaitons tester l'apport d'une meilleure classification entre les

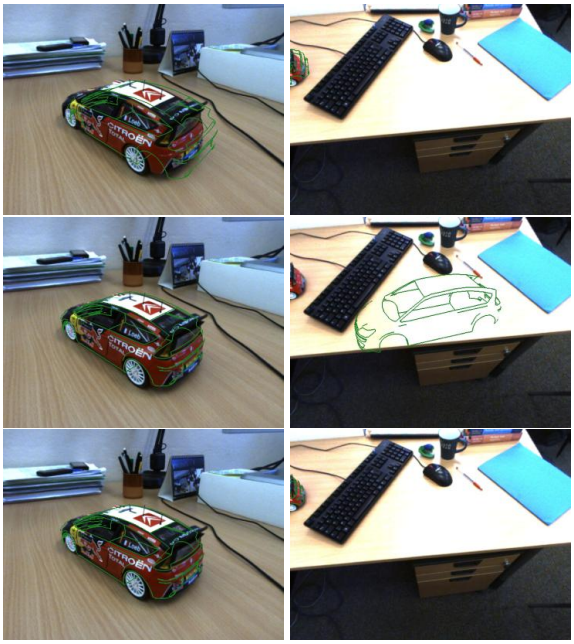

FIGURE 5 – Suivi d'objet 3D avec raffinement de type ajustement de faisceaux. En haut, au milieu, en bas : résultats obtenue respectivement avec les algorithmes de raffinement BA, BA_M, BA_M&E.

deux parties de la scène. Cette étape est actuellement effectuée par lancés de rayon. Nous pensons qu'une segmentation image peut aider à une classification plus précise.

Références

[1] Gabriele Bleser, Harald Wuest, and Didier Stricker. Online camera pose estimation in partially known and dynamic scenes. In *ISMAR*, 2006.

[2] Michela Farenzena, Adrien Bartoli, and Youcef Mezouar. Efficient camera smoothing in sequential structure-from-motion using approximate cross-validationt. In *ECCV*, 2008.

[3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[4] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *ISMAR*, 2007.

[5] Vincent Lepetit and Pascal Fua. Monocular model-based 3d tracking of rigid objects : A survey. In *FTCGV*, 2005.

[6] Pierre Lothe, Steve Bourgeois, Fabien Dekeyser, Eric Royer, and Michel Dhome. Towards geographical referencing of monocular slam reconstruction using 3d city models : Application to real-time accurate vision-based localization. In *CVPR*, 2009.

[7] D. Marquardt. An algorithm for least-squares estimation of non linear parameters. *J. Soc. Industr. Appl. Math.*, 11(1) :431–444, 1963.

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

[9] David Nister, Oleg Naroditsky, and James Bergen. Visual odometry. In *CVPR*, 2004.

[10] Eric Royer, Maxime Lhuillier, Michel Dhome, and Thierry Chateau. Localization in urban environments : Monocular vision compared to a differential gps sensor. In *CVPR*, 2005.

[11] Gilles Simon and Marie-Odile Berger. Pose estimation for planar structures. *IEEE Computer Graphics and Applications*, 22(6) :46–53, 2002.

[12] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *IJCV*, 80(2) :189–210, 2008.

[13] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCVW : International Workshop on Vision Algorithms Theory and Practice*, 2000.