SLAM contraint en environnement de grande taille

Datta Ramadasan, Marc Chevaldonné, Thierry Chateau

SLAM contraint en environnement de grande taille

D. Ramadasan[1] M. Chevaldonné[2] T. Chateau[1]

[1] Insitut Pascal UMR 6602 CNRS-UBP-IFMA
[2] ISIT UMR 6284 Uda-CNRS

datta.ramadasan@univ-bpclermont.fr

Résumé

Ce papier présente un algorithme de localisation et cartographie simultanée (ou SLAM pour Simultaneous Localization And Mapping) utilisant des contraintes issues d'un modèle d'environnement, on parle alors de SLAM contraint. L'objectif est d'utiliser un modèle 3D épars de grande taille issu d'une reconstruction de type SFM (Structure From Motion) afin d'estimer le mouvement de la caméra relativement au modèle, même lorsque ce dernier est peu ou pas observé. Il est nécesaire pour cela d'inclure des contraintes issues du modèle d'environnement dans l'algorithme de SLAM. Nous montrons que la cartographie simultanée permet l'utilisation d'un modèle approximatif ou partiellement obsolète avec une localisation plus fiable en conditions difficiles.

Mots Clef

SLAM temps réel, ajustement de faisceaux incrémental, contrainte planaire, navigation autonome.

Abstract

This paper presents an algorithm for simultaneous localization and mapping (or SLAM) using constraints from an environment model, this is called constrained SLAM. The aim is to use a large sparse 3D model coming from a Structure From Motion algorithm (SFM) to estimate the motion of the camera relatively to the model, even when this model is badly observed. To do so, it is necessary to include constraints from the environment model in the SLAM algorithm. We show that the simultaneous mapping allows the use of an approximate or partially obsolete model with a more reliable localization in difficult conditions.

Keywords

Real-time SLAM, incremental bundle adjustment, planar constraint, autonomous navigation.

1 Introduction

Les avancées en matière de SLAM *(Simultaneous Localization And Mapping)* monoculaire de la dernière décennie ont permis aux méthodes basées ajustement de faisceaux [10] d'atteindre des performances compatibles avec des applications temps réel. Des travaux plus récents utilisent des *a priori* sur la géométrie de la scène observée pour contraindre les paramètres du problème afin d'augmenter la robustesse et les performances de la méthode.

Nous utilisons une méthode de SLAM basée image clé semblable à celles dévelopées par Mouragnon *et al.* [6] et Klein *et al.* [3]. Ces méthodes sous-échantillonnent le flux vidéo en images clés (considérées pertinentes) telles que les différences entre chacune d'elles permettent l'estimation de la trajectoire caméra. L'originalité de ces méthodes réside dans l'optimisation locale qui ne remet en cause qu'un sous-ensemble de paramètres. Celui-ci correspond aux 3 dernières images clés dans [6] et aux 5 plus proches dans [3] avec les points 3D associés. Ces deux approches incluent également dans la fonction de coût les erreurs liées à la reprojection des points 3D vus dans des poses caméras non optimisées. Ceci a pour effet d'ancrer la partie plus ancienne de la trajectoire afin de limiter la dérive induite par l'accumulation d'erreur. Ces méthodes fonctionnent en environnement inconnu, mais il est souvent possible d'utiliser des *a priori*.

Plus récemment, Tamaazousti *et al.* proposent d'ajouter des contraintes sur les poses et les points 3D dans l'ajustement de faisceaux introduit dans [6]. Ces contraintes proviennent d'observations liées à un objet présent dans la scène et dont le modèle est connu sous forme d'un modèle CAO ou d'un nuage de points 3D épars. Dans [5], Lothe *et al.* utilise un modèle grossier d'environnement de grande taille issu d'un SIG (Système d'Information Géographique) pour appliquer des contraintes planaires sur les points 3D du SLAM et ainsi corriger la dérive lors d'un processus hors-ligne.

L'approche présentée ici reprend le travail de Tamaazousti [9], sur l'ajustement de faisceaux contraint par un nuage de point 3D éparse issu d'une reconstruction préalable, pour l'appliquer à un environnement de grande taille (trajectoires de plusieurs kilomètres) afin de corriger la dérive dans un processus temps réel.

2 Algorithme de SLAM contraint

La reconstruction incrémentale consiste à cartographier l'environnement à partir d'un flux vidéo à travers un processus itératif de sélection d'images clés et d'optimisation. Ce processus doit offrir un compromis satisfaisant entre la pré-

cision du résultat et le temps nécessaire pour effectuer les calculs. On considère le processus temps réel si le temps de mise à jour de la carte est inférieur au temps d'acquisition entre deux images clés. On utilise une caméra calibrée dans un environnement structuré et on suppose que le mouvement et la vitesse d'acquisition de la caméra permettent la mise en correspondance d'amers visuels entre deux images successives. La détection des points d'intérêt sur les images est réalisée avec le détecteur de Harris *et al.* [1], et leur appariement entre les images utilise le descripteur ZNCC *Zero Mean Normalized Cross Correlation* (qui a l'avantage d'être invariant à un changement affine de la luminosité).

2.1 Principe de la méthode

On dispose d'une reconstruction 3D éparse d'un environnement issue d'un algorithme de SFM ou plus particulièrement d'un algorithme de SLAM. Cette reconstruction préalable, que l'on nomme modèle de référence dans la suite du papier, est représentée par une trajectoire composée d'images clés, de points 3D et de points d'intérêts (composés d'une position 2D dans l'image et d'un descripteur) tel que chaque point 3D soit associé à un ensemble d'image clé (où il est observé) avec un descripteur différent dans chacune d'elles. Dans la méthode présentée, les paramètres du modèle de référence ne sont jamais remis en cause, ce qui se traduit par des paramètres ayant zéro degré de liberté. Le SLAM contraint prend en entrée la reconstruction de référence, et après une phase d'initialisation section 2.2, les positions successives de la caméra sont calculées en se basant à la fois sur les points 3D du modèle de référence et la carte du SLAM section 2.3. Lorsque le nombre de points 3D utilisés pour la localisation tend à diminuer, une nouvelle image clé est ajoutée à la carte du SLAM section 2.4. Un ajustement de faisceaux incrémental section 2.5 est alors utilisé pour optimiser la fin de la trajectoire du SLAM (poses et points 3D) mais sans remettre en cause les points 3D issu de la référence. Ceux-ci sont utilisés comme contraintes sur les poses du SLAM afin d'assurer une cohérence entre la cartographie temps réel et le modèle de référence.

2.2 Initialisation

La phase d'initialisation permet de déterminer la pose courante de la caméra par rapport à la trajectoire de référence sans utiliser d'*a priori* de position. L'initialisation est utilisée dans deux circonstances : au lancement du programme pour initier le SLAM, et par la suite lorsque le système a trop dérivé et qu'une ré-initialisation est nécessaire. La trajectoire de référence étant potientiellement de très grande dimension, nous utilisons la bibliothèque C++ FLANN [7] pour indexer l'ensemble des descripteurs (de la référence) dans une structure de type kD-tree en utilisant la ZNCC comme critère de distance. L'étape d'initialisation consiste alors à détecter l'ensemble des descripteurs visibles dans l'image courante afin d'effectuer une requête des 20 plus proches voisins sur l'arbre de descripteurs. La requête ainsi formulée renvoie pour chaque descripteur de l'image courante une liste de 20 descripteurs provenant de la référence. Un calcul de pose basé sur la méthode dite RANSAC (*Random Sample Consensus*) exploite ensuite les associations 2D-3D afin de sélectionner les appariements les plus pertinents parmi l'ensemble des candidats. La pose obtenue est alors ajoutée à la carte du SLAM comme image clé.

2.3 Calcul de pose à deux images clés

La localisation de la caméra s'effectue par mise en correspondance des amers visuels détectés dans l'image courante avec deux images clés : une image clé de la référence et une image clé du SLAM. L'image clé de la référence est la plus proche (en terme de distance et avec une orientation similaire) de la dernière position connue de la caméra. L'image clé du SLAM est la plus récente. Lorsqu'un descripteur de l'image courante est semblable (selon le critère de ZNCC et en supposant les points de vues proches) à la fois à un descripteur de la référence et à un descripteur du SLAM, le descripteur de la référence est préféré. Cette gestion des appariement est nécessaire car les points provenant du SLAM ont une apparence très proche des descripteurs courants (conditions d'observations quasi-identique) et seraient trop souvent préférés à ceux de la référence. Un RANSAC est effectué sur la liste d'appariement obtenue. Chaque itération du RANSAC consiste en un tirage aléatoire de 3 points tirés parmi l'ensemble des appariement (référence et SLAM confondu). Les 3 points sont utilisés pour calculer une pose avec l'algorithme P3P [4] dont la pertinence est évaluée en comptant le nombre de point 3D (provenant à la fois de la référence et du SLAM) étant catégorisé inliers *i.e* dont l'erreur de reprojection est inférieur à un seuil (en pratique, on utilise 2 pixels). La pose maximisant le nombre d'inlier est conservée et raffinée ensuite avec un algorithme de minimisation de type Gauss-Newton en utilisant l'ensemble des points inliers.

2.4 Ajout d'image clé

Pour sélectionner les images clés du SLAM, nous définissons un critère tel qu'une image correctement localisée soit ajoutée à la carte courante comme image clé si l'image suivante possède moins de 20% des points d'intérêt correctement associés à des points 3D déjà observés 3 fois dans la référence et/ou la carte du SLAM. Ce critère offre les garantis suivantes :

- l'image clé contient au moins 20% de points en commun avec la carte et/ou la référénce,
- la carte contient des points 3D observés au minimum 3 fois,
- le nombre de points propagés est proportionnel au nombre de points détectés,
- aucune image clé n'est ajoutée tant que l'on retrouve suffisamment de point dans la référence.

2.5 Ajustement de faisceaux incrémental contraint

Lorsque les points 3D de la référence et de la carte courante ne sont plus suffisants pour satisfaire le critère défini dans

la section 2.4, une image clé est ajoutée à la carte en utilisant la pose issue de la localisation section 2.3. Les points 3D catégorisés inlier lors du calcul de pose sont associés à cette image clé ainsi que leurs descripteurs correspondants. L'ajout de ces données est susceptible de remettre en cause les paramètres déjà présents dans la carte, c'est pourquoi il est nécessaire d'effectuer une optimisation des paramètres afin de maintenir la cohérence entre les données. Toutefois, pour effectuer un minimum de calcul, on restreint les paramètres à optimiser à un sous-ensemble directement remis en cause par l'ajout des nouvelles informations. Pour sélectionner les paramètres à optimiser, Klein *et al.* [3] définissent l'ensemble C composé de la dernière image clé ainsi que des 4 images clés les plus proches. Ils définissent également l'ensemble P contenant l'ensemble des points 3D vus dans C. Ils optimisent alors les poses clés de C ainsi que les points 3D de P en tenant compte des projections des points 3D de P dans toutes les images où ils apparaissent. Nous proposons de redéfinir C comme l'ensemble des images clés (de la carte courante) ayant des points 3D en commun avec la dernière image clé. Cela permet de sélectionner les poses les plus pertinentes car directement remises en question par les nouvelles observations. Toutefois, on limite le nombre d'images clés de C à 10 pour conserver une résolution en temps réel du problème. De plus, les paramètres de la référence ne sont pas remis en cause. Ainsi, les ensembles C et P ne contiennent aucun paramètre de la référence. Cela signifie que les observations des points 3D provenant de la référence dans les images clés sont utilisées dans la fonction d'erreur mais les paramètres des points 3D sont constants. Les ensembles de poses et de points à ne pas optimiser sont notés respectivement $\overline{C}$ et $\overline{P}$. La fonction de coût $\mathcal{E}$ à minimiser se compose de la somme de trois termes, avec π la fonction d'erreur de reprojection correspondant à la distance pixelique signée entre la projection d'un point 3D dans une image et l'observation 2D associée :

$$\mathcal{E} = \sum \|\pi(C,P)\|^2 + \sum \|\pi(\overline{C},P)\|^2 + \sum \|\pi(C,\overline{P})\|^2 \quad (1)$$

Les dérivées des paramètres non optimisés étant nulles, la jacobienne J de la fonction de coût de l'ajustement de faisceaux contraint s'écrit :

$$J = \begin{bmatrix} J_C^{\pi(C,P)} & J_P^{\pi(C,P)} \\ 0 & J_P^{\pi(\overline{C},P)} \\ J_C^{\pi(C,\overline{P})} & 0 \end{bmatrix} \quad (2)$$

avec J_X^F la jacobienne de la fonction F en fonction des paramètres de X.

2.6 Résolution des équations normales

L'ensemble X des paramètres de poses et points 3D sont optimisés en utilisant l'algorithme de moindre carré non-linéaire Levenberg-Marquardt [2]. A chaque itération i, les paramètres sont mis à jour :

$$X_{i+1} = X_i - (J_i^T J_i + I\lambda_i)^{-1} J_i^T \epsilon_i \quad (3)$$

avec J la jacobienne de la fonction de coût, ϵ l'ensemble des résidus et λ un paramètre d'amortissement. Le calcul de l'incrément Δ à appliquer aux paramètres X pour diminuer l'erreur de la fonction de coût revient à résoudre le système linéaire (ou équations normales) suivant :

$$(J_i^T J_i + I\lambda_i)\Delta_i = J_i^T \epsilon_i \quad (4)$$

La résolution efficace des équations normales en utilisant les *a priori* sur la structure éparse de la jacobienne est détaillée dans [6].

2.7 Performances de l'implémentation

L'implémentation (en langage C++) de l'algorithme de SLAM contraint présentée ici offre des performances temps réel. Le temps d'exécution est d'environ 40 ms de la détection des points d'intérêts à l'optimisation incrémental, sur un ordinateur de bureau équipé d'un processeur i7 3.4Ghz en utilisant un seul fil d'exécution (processus système). La détection de points d'intérêt se fait sur des images de résolution 752×480 et 700 points d'intérêt sont détectés. Les temps moyen nécessaire à chaque étape du processus sont, 4 ms pour la détection des points d'intérêt, 3 ms pour l'appariement, 24 ms pour le RANSAC et le calcul de pose , et 9 ms pour l'ajustement de faisceaux incrémental.

3 Expérimentation

3.1 Méthodologie

L'expérimentation présentée ci-dessous compare le SLAM contraint à la méthode de localisation utilisée par Royer *et al.* [8]. La trajectoire de référence est enregistrée sous des conditions météorologiques idéales avec une caméra *global shutter* de résolution 752*480 et une fréquence d'acquisition de 20 hz. La figure 1 est un échantillon des images obtenues. La distance parcourue durant l'acquisition, de 2.2km, constitue une séquence de 9000 images. La reconstruction 3D est effectuée par une algorithme de SLAM classique dont le résultat est une trajectoire constituée de 1203 images clés, 83652 points 3D et 1465069 points d'intérêt. Cette reconstruction constitue la trajectoire de référence de l'expérimentation visible sur la figure 2. Cette figure permet de comparer visuellement la trajectoire réellement effectuée avec le résultat de la reconstruction. On remarque une légère dérive de la trajectoire de référence, mais cela ne remet par en cause l'expérimentation car l'objectif est l'évaluation des méthodes de localisation par rapport à un modèle de référence, même si celui-ci est approximatif.
Une deuxième séquence d'images est enregistrée sous la pluie (avec les essuie-glaces du véhicule allumés) quelques jours après la première séquence et suit le même parcours. Ce second enregistrement est constitué de 6100 images acquisent avec la même caméra. Le nombre d'image est inférieur à la première séquence car la conduite du véhicule était plus rapide. La figure 3 est un échantillon des images constituant cet enregistrement (les images sont prises au même endroit du parcours par rapport à la figure 1).

FIGURE 1 – Echantillon d'images issus du premier enregistrement destiné à la reconstruction du modèle de référence.

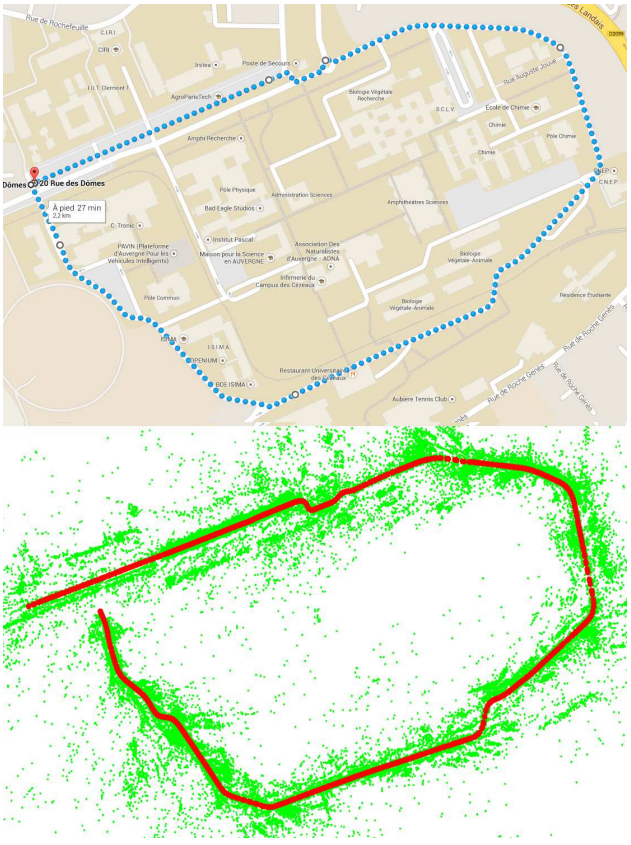

FIGURE 2 – La première image est la trajectoire réellement effectuée visualisée sur une carte de Google Map. La deuxième image est une vue de dessus de la trajectoire issue de la reconstruction SLAM.

FIGURE 3 – Echantillon d'images issus du second enregistrement destiné à l'évaluation des méthodes de localisation. Les images sont sélectionnées approximativement au même endroit du parcours que les images de la figure 1.

3.2 Test de robustesse

Le test consiste à localiser les images de la deuxième séquence en utilisant la trajectoire de référence comme modèle (issue de la reconstruction de la première séquence). La méthode de localisation de Royer, que l'on note Loc, est comparée à la méthode de localisation par SLAM contraint que l'on note $SLAMC$. Plusieurs répétitions du tests sont effectuées en appliquant un mécanisme d'appauvrissement de la trajectoire consistant à supprimer à chaque itération 5% des points 3D (tirés aléatoirement) du modèle de référence. Pour évaluer la robustesse des méthodes, on compte le nombre de fois où les calculs de pose ont obtenu moins de 10 points classifiés inliers. Autrement dit, si la pose est calculée avec moins de 10 points 3D observés (avec une erreur de reprojection dans l'image courante inférieur à 2 pixels), alors la localisation est un échec. En pratique, la méthode $SLAMC$ obtient toujours un nombre important de points grâce à la cartographie simultanée. Toutefois, si celle-ci ne retrouve plus de point provenant de la référence, il n'est pas possible de savoir si la position calculée est fiable dans le repère de la référence. C'est pourquoi on considère que la localisation de la méthode $SLAMC$ échoue si aucun point de la référence n'est utilisé. La figure 4 illustre le résultat de ce test. Jusqu'à 75% d'appauvrissement de la trajectoire de référence, la méthode Loc a un taux d'échec inférieur à 2%, et augmente rapidement au delà. La méthode $SLAMC$ quant à elle subit moins de 0.5% d'échec jusqu'à 85% d'appauvrissement.

3.3 Comportement de l'ajustement de faisceaux contraint

Comparativement à une méthode de SLAM classique, le SLAM contraint inclut des observations supplémentaires dans la fonction de coût. On pourrait supposer que le temps nécessaire au processus d'optimisation est alors plus important. Or, le SLAM contraint tel qu'il a été décrit précédemment utilise conjointement des points 3D provenant de la cartographie temps réelle (points 3D à 3 degrés de

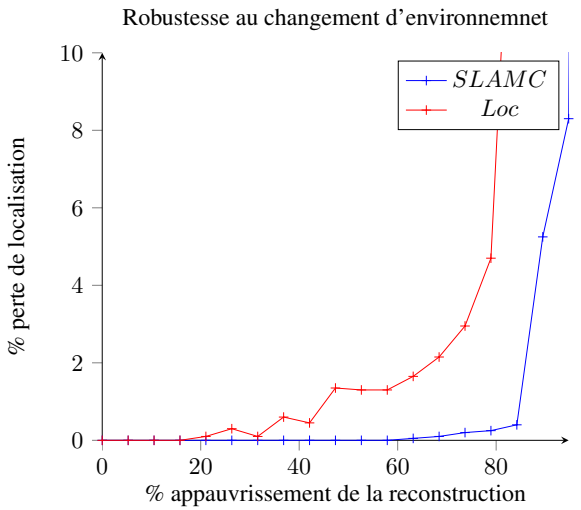

FIGURE 4 – % pourcentage d'échec de localisation des méthodes $SLAMC$ et Loc en fonction de l'appauvrissement des données du modèle de référence.

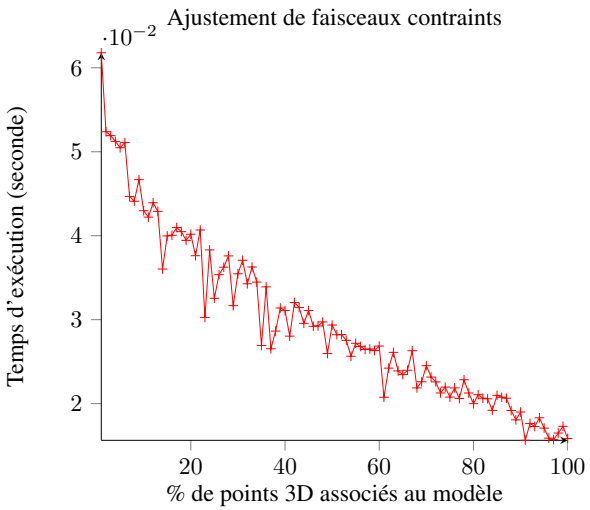

FIGURE 5 – Temps d'exécution de l'ajustement de faisceaux contraint en fonction du nombre de points 3D associés au modèle.

liberté) et du modèle de référence (points 3D à 0 degré de liberté). Les points 3D du modèle n'étant pas remis en cause, ils ne constituent pas de paramètres supplémentaires dans l'optimisation. La complexité algorithmique de la résolution des équations normales étant en grande partie liée au nombre de paramètres et non au nombre d'observations, l'ajustement de faisceaux contraint est effectué dans un temps inférieur à l'ajustement de faisceaux classique. La figure 5 illustre le temps nécessaire, avec l'algorithme de Levenberg-Marquardt, pour la minimisation de la fonction de coût en fonction du nombre de points 3D provenant du modèle. Ce test est réalisé sur des données de synthèse. Pour chaque occurence du test, le problème est composé de 10 poses et 1000 points 3D répartis entre le modèle de référence et la carte issue du SLAM en fonction du pourcentage d'association au modèle (le nombre d'observation total est constant quelque soit la répartition des points 3D). On observe que le temps d'exécution est d'autant plus faible que la fonction de coût fait intervenir une proportion importante de points 3D provenant du modèle de référence donc ayant 0 degré de liberté. Entre 1% de points 3D associés au modèle et 100%, le temps d'exécution est divisé par 4.

4 Conclusion

L'utilisation du SLAM contraint permet d'augmenter la robustesse de la localisation face aux changements importants de l'environnement en utilisant le processus de cartographie temps réel pour combler le manque de points provenant du modèle de référence. Le test d'ajustement de faisceaux contraint sur données de synthèse montre que l'utilisation de paramètres à 0 degré de liberté permet un gain en temps de calcul allant jusqu'à un facteur 4 pour l'optimisation du problème, comparativement au SLAM classique. De plus, les tests effectués sur une séquence réelle en environnement extérieur, dans des conditions de rejeu différentes, ont montré une robustesse accrue de la méthode de SLAM contraint comparativement à une localisation basée modèle. Ainsi, la méthode sous contrainte permet une localisation malgré un appauvrissement de 80% du modèle de référence.

Références

[1] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*. Manchester, UK, 1988.

[2] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000.

[3] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, ISMAR*. IEEE, 2007.

[4] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2969–2976. IEEE, 2011.

[5] P. Lothe, S. Bourgeois, E. Royer, M. Dhome, and S. Naudet-Collette. Real-time vehicle global localisation with a single camera in dense urban areas : Exploitation of coarse 3d city models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 863–870. IEEE, 2010.

[6] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3d reconstruction. In *Computer Vision and Pattern Recognition*. IEEE, 2006.

[7] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In

International Conference on Computer Vision Theory and Application VISSAPP'09), pages 331–340. INSTICC Press, 2009.

[8] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3) :237–260, 2007.

[9] M. Tamaazousti. *L'ajustement de faisceaux contraint comme cadre d'unification des méthodes de localisation : application à la réalité augmentée sur des objets 3D*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2013.

[10] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment, a modern synthesis. In *Vision algorithms : theory and practice*. 2000.