Fusion GPS-vision à l'aide d'ajustements de faisceaux contraints

Maxime Lhuillier
LASMEA UMR 6602, UBP/CNRS, 24 avenue des Landais, 63177 Aubière Cedex
http ://maxime.lhuillier.free.fr

Résumé

Il y a deux problèmes lorsque l'on applique un ajustement de faisceaux (AF) sur des séquences d'images longues : le temps de calcul et la dérive (i.e. l'accumulation d'erreurs). Une méthode récente réduit le temps de calcul grâce à un AF local appliqué incrémentalement. De plus, la fusion avec des données GPS devrait permettre de réduire la dérive. Une méthode de fusion existante est un AF minimisant une somme pondérée d'erreurs images et GPS.
Cet article introduit deux AFs contraints pour la fusion, qui imposent une limite supérieure sur l'erreur de reprojection. Ces AFs sont des alternatives à l'AF de fusion existant, ce dernier ne garantissant pas une faible erreur de reprojection et nécessitant la donnée d'une pondération. Ensuite, les trois AFs de fusion sont intégrés dans un système de reconstruction visuelle incrémental basé sur l'AF local. Enfin, des résultats de fusion avec un GPS bas coût sont comparés sur une longue séquence d'images monoculaires.

Mots Clef

Ajustement de faisceaux local, fusion, GPS bas coût

Abstract

Two problems occur when bundle adjustment (BA) is applied on long image sequences : the large calculation time and the drift (or error accumulation). In recent work, the calculation time is reduced by local BAs applied in an incremental scheme. The drift may be reduced by fusion of GPS and Structure-from-Motion. An existing fusion method is BA minimizing a weighted sum of image and GPS errors. This paper introduces two constrained BAs for fusion, which enforce an upper bound for the reprojection error. These BAs are alternatives to the existing fusion BA, which does not guarantee a small reprojection error and requires a weight as input. Then the three fusion BAs are integrated in an incremental Structure-from-Motion method based on local BA. Lastly, we will compare the fusion results on a long monocular image sequence and a low cost GPS.

Keywords

Local Bundle adjustment, Fusion, low cost GPS

1 Introduction

L'ajustement de faisceaux (AF) est une méthode itérative d'estimation des poses de caméra et des points 3D détectés dans une séquence d'images. Les points et poses obtenus minimisent la somme des carrés des erreurs de reprojection. Bien que l'AF soit connu depuis longtemps [13], c'est toujours un sujet de recherches actives.

Des développements récents concernent l'accéleration pour les longues séquences. Une méthode de gradient conjugué peut être plus rapide qu'une méthode de factorisation (avec ré-ordonnancement des variables) pour résoudre le système réduit des caméras [5]. Des AFs locaux [11] (AFL) sont appliqués plusieurs fois pour raffiner tous les paramètres 3D des images clefs les plus récentes d'une vidéo. Les précisions et incertitudes des poses calculées sont similaires à celles de l'AF global appliqué sur toute la vidéo [11, 3].

Un autre sujet de recherche sur l'AF est la fusion de données provenant de différents capteurs. La fusion est utile pour réduire l'accumulation d'erreurs de la reconstruction visuelle (ou SfM pour "structure-from-motion"), inévitable pour une longue séquence d'images (en particulier lorsque la caméra est monoculaire). L'AF global est utilisé en photogrammétrie aérienne pour combiner des erreurs images, GPS et inertielles pondérées par leurs covariances [9]. On peut aussi essayer d'inclure les pseudo-distances du GPS directement comme mesures dans l'AF [2]. Dans un contexte différent, les erreurs de reprojection des points 3D impliqués dans l'AF sont modifiées de sorte que les points sont contraints dans des plans verticaux stockés dans une base de donnée GIS [8]. Des travaux récents combinent des mesures GPS et images [6] (ou inertielles et images [10]) à l'aide d'un AFL, qui minimise une somme pondérée d'erreurs GPS (ou inertielles) et images. Dans [10], plusieurs méthodes de pondération sont proposées. Dans [6], les expériences sont limitées à une séquence courte (70m) et l'erreur GPS est définie par un polynôme de degrés élevé.

Cet article (version améliorée de [7]) a plusieurs contributions. Premièrement, la partie 2 résume brièvement 5 AFs qui peuvent résoudre le problème de la fusion GPS-vision. Seules des méthodes efficaces du second ordre dérivant de la méthode de Levenberg-Marquardt (LM) sont présentées ici. Deuxièmement, la partie 3 introduit les deux nouveaux AFs pour la fusion. L'un implique une contrainte d'inégalité et l'autre une contrainte d'égalité. Ces AFs imposent une limite supérieure sur l'erreur de reprojection, alors que les AFs existants de fusion [9, 10, 6] ne garantissent pas une erreur de reprojection faible et nécessitent une pondération. Troisièmement, la partie 4 donne des informations techniques importantes (algorithmes détaillés, condi-

tions pratiques pour faciliter la fusion). Enfin, la partie 5 compare les résultats des AFs de fusion dans un contexte utile pour les applications : le SfM incrémental basé sur l'AFL [11]. Dans les expériences, un GPS bas coût et une caméra monoculaire calibrée sont fixés sur un véhicule qui se déplace dans une zone urbaine.

2 AFs pour la fusion GPS-vision

2.1 Principales notations et hypothèses

La norme euclidienne est $||.||$. Des fontes différentes sont utilisées pour les vecteurs (ex. : $\mathbf{x}$), les matrices (ex. : $\mathtt{H}$) et les fonctions/réels (ex. : e). Supposons que $\mathtt{H} > 0$, i.e. $\mathtt{H}$ est symétrique définie positive. Soient $\mathtt{D}$ un bloc diagonal de $\mathtt{H}$ et $\mathrm{diag}(\mathtt{H})$ la matrice diagonale obtenue en forcant à 0 les coefficients non diagonaux de $\mathtt{H}$. On rappelle que l'on a $\mathtt{D} > 0$, $\mathrm{diag}(\mathtt{H}) > 0$, $\mathtt{H} + \mathtt{A}^T\mathtt{A} > 0$, et $\mathtt{H}^{-1} > 0$.
Soient $\mathbf{x}$ le vecteur des paramètres 3D (poses de caméra et points 3D) et $e(\mathbf{x})$ la somme des carrés des erreurs de reprojection de $\mathbf{x}$. Dans cet article, on suppose que le $\mathbf{x}$ de départ/d'entrée des AFs de fusion est le minimiseur $\mathbf{x}^*$ de e, i.e. $\forall \mathbf{x}, e(\mathbf{x}^*) \leq e(\mathbf{x})$, qui est donné par un AF standard. Soit $\mathbf{x}_1$ un vecteur de position(s) de la caméra. L'ordre des variables est tel que $\mathbf{x}^T = \begin{pmatrix}\mathbf{x}_1^T & \mathbf{x}_2^T\end{pmatrix}$. On note $\mathtt{P}$ tel que $\mathbf{x}_1 = \mathtt{P}\begin{pmatrix}\mathbf{x}_1^T & \mathbf{x}_2^T\end{pmatrix}^T$ et $e(\mathbf{x}_1, \mathbf{x}_2) = e(\begin{pmatrix}\mathbf{x}_1^T & \mathbf{x}_2^T\end{pmatrix}^T)$.
Soit $\mathbf{x}_1^{gps}$ un vecteur de position(s) fourni(s) par le GPS au(x) même(s) instant(s) que $\mathbf{x}_1$. En supposant que la dérive du GPS est bornée et que celle du SfM est non bornée, la sortie idéale $\mathbf{x}$ des AFs de fusion est telle que $\mathbf{x}_1 \approx \mathbf{x}_1^{gps}$. Soit e_t un seuil qui est légèrement plus grand que le minimum $e(\mathbf{x}^*)$ de e. Dans notre contexte, le $\mathbf{x}$ final/de sortie des AFs de fusion est supposé être acceptable si son erreur de reprojection est similaire à $e(\mathbf{x}^*)$, i.e. si

$$e(\mathbf{x}) < e_t. \tag{1}$$

Cette inégalité définit une région d'indifférence pour $\mathbf{x}$ [1].

2.2 AFs sans contrainte explicite

Ces AFs ont été utilisés [9] pour combiner des mesures provenant de différent capteurs. On les appelle ici "AF non contraints". Une première approche simple est l'AF qui minimise $\mathbf{x}_2 \mapsto e(\mathbf{x}_1^{gps}, \mathbf{x}_2)$ après avoir remplacé le $\mathbf{x}_1$ de départ by $\mathbf{x}_1^{gps}$. Cet AF peut converger si la valeur de départ de $\mathbf{x}_1 - \mathbf{x}_1^{gps}$ est suffisamment petite (ce qui est hasardeux dans notre cas).
Une seconde approche est la minimisation d'une somme d'erreurs pondérées des deux capteurs : on minimise

$$e_U(\mathbf{x}) = e(\mathbf{x}) + \beta||\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}||^2. \tag{2}$$

Ici, les problèmes sont le choix adéquat du poids β et le risque de perte d'inlier due au terme $\beta||\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}||^2$. Un "inlier" est un point détecté dans une image et impliqué dans e tel que l'erreur de reprojection associée est inférieure à un seuil. Remplacer $\beta||.||^2$ par une forme quadratique définie par une matrice de covariance n'élimine pas ces problèmes.

2.3 AFs avec contrainte d'égalité

Deux méthodes sont possibles [13] : une méthode de programmation quadratique séquencielle (PQS) et une méthode de réduction. Les deux minimisent $e(\mathbf{x})$ sous la contrainte $c_E(\mathbf{x}) = 0$. Une itération de ces AFs contraints améliore $\mathbf{x}$ en additionnant un pas $\mathbf{\Delta}$ satisfaisant la contrainte linéarisée $c_E(\mathbf{x} + \mathbf{\Delta}) \approx c_E(\mathbf{x}) + \frac{\partial c_E}{\partial \mathbf{x}}\mathbf{\Delta} = \mathbf{0}$. Comme pour un AF non contraint, un amortissement (damping) est utilisé pour définir $\mathbf{\Delta}$ entre le pas de Gauss-Newton (qui minimise le développement de Taylor à l'ordre 2 de e) et un pas de descente de gradient. Les développements de Taylor supposent que $\mathbf{\Delta}$ est suffisamment petit, ce qui nécessite donc une valeur suffisamment petite de $c_E(\mathbf{x}) = -\frac{\partial c_E}{\partial \mathbf{x}}\mathbf{\Delta}$.
PQS estime simultanément $\mathbf{\Delta}$ et les multiplicateurs de Lagrange pour c_E. La méthode de réduction s'applique si $\frac{\partial c_E}{\partial \mathbf{x}_1}$ est carrée et inversible. Dans ce dernier cas, le pas $\mathbf{\Delta}_1$ de $\mathbf{x}_1$ s'exprime en fonction du pas $\mathbf{\Delta}_2$ de $\mathbf{x}_2$, et le développement de Taylor de e ne dépend que de $\mathbf{\Delta}_2$.

2.4 AF avec contrainte d'inégalité

Il existe une autre méthode, qui utilise une fonction de pénalité [1]. Dans notre contexte, les itérations de cet AF contraint impose la contrainte de l'éq. 1, i.e. $c_I(\mathbf{x}) > 0$ avec $c_I(\mathbf{x}) = e_t - e(\mathbf{x})$. Il s'agit de minimiser

$$e_I(\mathbf{x}) = \gamma/c_I(\mathbf{x}) + ||\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}||^2 \tag{3}$$

avec $\gamma > 0$. Ici, $\mathbf{x} \mapsto \mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}$ est minimisée tandis que la fonction de pénalité $\gamma/c_I(\mathbf{x})$ impose la contrainte d'inégalité. Cette pénalité est le terme principal (arbitrairement grand) dans le voisinage de $c_I(\mathbf{x}) = 0$, et elle ne perturbe pas trop les minimiseurs de $\mathbf{x} \mapsto ||\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}||^2$ ailleurs.

2.5 AFs contraints proposés

UBA ("Unconstrained Bundle Adjustment") est l'AF qui minimise e_U (éq. 2). On introduit ici deux méthodes de fusion **IBA** et **EBA**. IBA est l'AF avec contrainte d'inégalité qui minimise e_I (éq. 3). A notre connaissance, bien que le principe soit simple, IBA n'avait pas été utilisé pour fusionner un résultat de vision avec un autre capteur (GPS ou autre). EBA est un AF avec contrainte d'égalité, qui est dérivé de la méthode de réduction (partie 2.3). Cette dérivation n'est pas si simple : d'un coté l'AF original nécessite un petit $||c_E(\mathbf{x})||$, de l'autre $||c_E(\mathbf{x})|| = ||\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}||$ peut être grand. Ce problème est le même avec PQS.
Gardons en mémoire que les méthodes de fusion ont les propriétés suivantes. Le $\mathbf{x}$ d'entrée minimise e, i.e. $\mathbf{x} = \mathbf{x}^*$. Le $\mathbf{x}$ de sortie à une erreur de reprojection petite, i.e. $e(\mathbf{x}) < e_t$ (EBA doit être conçu pour satisfaire cette contrainte ; la sortie de UBA est ignorée si $e(\mathbf{x}) > e_t$). Enfin, le sous-vecteur $\mathbf{x}_1$ du $\mathbf{x}$ de sortie est aussi proche que possible de $\mathbf{x}_1^{gps}$.

3 Itérations des AFs

La partie 3 décrit les itérations de LM, IBA et EBA (le premier est utile pour expliquer les deux autres). Dans tous

les cas, on vérifie qu'une itération réussie est possible.
Le développement de Taylor à l'ordre 2 de e est

$$e(\mathbf{x}+\mathbf{\Delta}) \approx e(\mathbf{x}) + \mathbf{g}^T\mathbf{\Delta} + 0.5\mathbf{\Delta}^T\mathtt{H}\mathbf{\Delta} \quad (4)$$

avec le gradient $\mathbf{g}$ et la hessienne $\mathtt{H}$ de e. La fonction de projection $E : \mathbb{R}^n \to \mathbb{R}^m$ satisfait $e(\mathbf{x}) = ||E(\mathbf{x})||^2$. Soit $\mathtt{J}$ la jacobienne de E en $\mathbf{x}$. On a $\mathbf{g} = 2\mathtt{J}^T E(\mathbf{x})$, l'approximation de Gauss-Newton $\mathtt{H} \approx 2\mathtt{J}^T\mathtt{J}$. On suppose $\mathtt{H} > 0$.

3.1 Levenberg-Marquardt sans contrainte

L'itération de LM qui minimise $e(\mathbf{x})$ sans contrainte est définie de la façon suivante (UBA minimise une fonction différente avec LM). Une méthode adéquate [4] est utilisée pour résoudre $(\mathtt{H} + \lambda\text{diag}(\mathtt{H}))\mathbf{\Delta} = -\mathbf{g}$ pour la valeur courante de $\mathbf{x}$ et un coefficient d'amortissement donné $\lambda > 0$. Si $e(\mathbf{x}+\mathbf{\Delta}) < e(\mathbf{x})$, l'itération a réussi : on remplace $\mathbf{x}$ par $\mathbf{x}+\mathbf{\Delta}$ et λ par $\lambda/10$. Sinon, on remplace λ par 10λ.

Itération réussie Supposons que toutes les itérations échouent et que $\mathbf{g} \neq \mathbf{0}$ (on ne part pas d'un minimum local). Dans ce cas, λ croit sans cesse. Un grand λ implique $\mathbf{\Delta} \approx \mathbf{\Delta}_d$ tel que $\lambda\text{diag}(\mathtt{H})\mathbf{\Delta}_d = -\mathbf{g}$. Plus λ est grand, plus $\mathbf{\Delta}$ est petit, et meilleure est l'approximation de Taylor de e. De plus, $\mathtt{H} > 0$ implique $(\mathtt{H} + \lambda\text{diag}(\mathtt{H}))^{-1} > 0$. Maintenant, l'approximation de Taylor à l'ordre 1 de e et $\mathbf{g} \neq \mathbf{0}$ impliquent

$$e(\mathbf{x}+\mathbf{\Delta}) \approx e(\mathbf{x}) - \mathbf{g}^T(\mathtt{H} + \lambda\text{diag}(\mathtt{H}))^{-1}\mathbf{g} < e(\mathbf{x}). \quad (5)$$

On obtient une itération réussie.

3.2 AF avec contrainte d'inégalité (IBA)

La méthode est la même que dans la partie 3.1, sauf pour le calcul de $\mathbf{\Delta}$. Soient x_i un coefficient de $\mathbf{x}$ et $f(\mathbf{x}) = \gamma/(e_t - e(\mathbf{x}))$. On a

$$\begin{aligned}\frac{\partial f}{\partial x_i} &= \frac{\gamma}{(e_t-e)^2}\frac{\partial e}{\partial x_i} \\ \frac{\partial^2 f}{\partial x_i \partial x_j} &= \frac{\gamma}{(e_t-e)^3}((e_t-e)\frac{\partial^2 e}{\partial x_i \partial x_j} + 2\frac{\partial e}{\partial x_i}\frac{\partial e}{\partial x_j}). \quad (6)\end{aligned}$$

On utilise alors l'approximation de Gauss-Newton $\mathtt{H} \approx 2\mathtt{J}^T\mathtt{J}$ et on obtient le gradient et la hessienne de e_I :

$$\begin{aligned}\mathbf{g}_I &= \frac{\gamma}{(e_t-e)^2}\mathbf{g} + 2\mathtt{P}^T(\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}) \\ \mathtt{H}_I &\approx \frac{2\gamma}{(e_t-e)^3}((e_t-e)\mathtt{J}^T\mathtt{J} + \mathbf{g}\mathbf{g}^T) + 2\mathtt{P}^T\mathtt{P}. \quad (7)\end{aligned}$$

Maintenant, on résout le système linéaire $(\mathtt{H}_I + \lambda\text{diag}(\mathtt{H}_I))\mathbf{\Delta} = -\mathbf{g}_I$. Celui-ci ne peut être résolu comme dans le cas non contraint car $\mathtt{H}_I$ n'est pas une matrice creuse à cause du terme dense $\mathbf{g}\mathbf{g}^T$. La partie 4.1 décrit une méthode efficace pour résoudre ce système.

Itération réussie Finalement, on vérifie que la décroissance de e_I est possible. L'éq. 7, l'éq. 1 et $\mathtt{J}^T\mathtt{J} > 0$ impliquent $\mathtt{H}_I > 0$. La décroissance de e_I se montre ensuite comme la décroissance de e dans la partie 3.1 (remplacer e par e_I, $\mathtt{H}$ par $\mathtt{H}_I$ et $\mathbf{g}$ par $\mathbf{g}_I$).

3.3 AF original avec contrainte d'égalité

On décrit maintenant l'itération de LM pour minimiser $e(\mathbf{x})$ sous la contrainte $c(\mathbf{x}) = 0$ dans sa version orinale [13]. On utilise les notations par blocs

$$\begin{pmatrix}\mathbf{x}_1 & \mathbf{\Delta}_1 & \mathbf{g}_1 \\ \mathbf{x}_2 & \mathbf{\Delta}_2 & \mathbf{g}_2\end{pmatrix} = \begin{pmatrix}\mathbf{x} & \mathbf{\Delta} & \mathbf{g}\end{pmatrix}, \begin{pmatrix}\mathtt{H}_1 & \mathtt{H}_{12} \\ \mathtt{H}_{21} & \mathtt{H}_2\end{pmatrix} = \mathtt{H} \quad (8)$$

et la jacobienne $\begin{pmatrix}\mathtt{C}_1 & \mathtt{C}_2\end{pmatrix}$ de c en $\mathbf{x}$. Dans notre cas, $\mathtt{C}_1 = \mathtt{I}$, $\mathtt{C}_2 = \mathtt{0}$ et le pas $\mathbf{\Delta}$ vérifie

$$c(\mathbf{x}+\mathbf{\Delta}) \approx c(\mathbf{x}) + \mathtt{C}_1\mathbf{\Delta}_1 + \mathtt{C}_2\mathbf{\Delta}_2 = c(\mathbf{x}) + \mathbf{\Delta}_1 = \mathbf{0}. \quad (9)$$

Donc $\mathbf{\Delta}$ est une fonction de $\mathbf{\Delta}_2$:

$$\Delta(\mathbf{\Delta}_2) = \begin{pmatrix}-c(\mathbf{x})^T & \mathbf{\Delta}_2^T\end{pmatrix}^T. \quad (10)$$

Grâce à l'éq. 4 et $\mathbf{\Delta} = \Delta(\mathbf{\Delta}_2)$, on obtient

$$e(\mathbf{x}+\Delta(\mathbf{\Delta}_2)) \approx \bar{e}_2 + \mathbf{\Delta}_2^T\bar{\mathbf{g}}_2 + 0.5\mathbf{\Delta}_2^T\mathtt{H}_2\mathbf{\Delta}_2 \quad (11)$$

avec

$$\begin{aligned}\bar{e}_2 &= e(\mathbf{x}) - \mathbf{g}_1^T c(\mathbf{x}) + 0.5c(\mathbf{x})^T\mathtt{H}_1 c(\mathbf{x}) \\ \bar{\mathbf{g}}_2 &= \mathbf{g}_2 - \mathtt{H}_{21}c(\mathbf{x}) \quad (12)\end{aligned}$$

Le pas $\mathbf{\Delta}_2$ vérifie $(\mathtt{H}_2 + \lambda\text{diag}(\mathtt{H}_2))\mathbf{\Delta}_2 = -\bar{\mathbf{g}}_2$. L'itération est maintenant celle de la partie 3.1 avec $\mathbf{\Delta} = \Delta(\mathbf{\Delta}_2)$.

3.4 AF avec contrainte d'égalité (EBA)

Modifier l'AF original Supposons que EBA soit l'AF original ci-dessus avec $c(\mathbf{x}) = \mathbf{x}_1 - \mathbf{x}_1^{gps}$ (on note maintenant $c = c_E$). Un premier problème est la condition de descente $e(\mathbf{x}+\Delta(\mathbf{\Delta}_2)) < e(\mathbf{x})$ pour tester le pas $\mathbf{\Delta} = \Delta(\mathbf{\Delta}_2)$. Dans notre contexte de fusion, la valeur initiale de $\mathbf{x}$ est $\mathbf{x}^*$, qui minimise e. Donc la condition de descente n'est jamais satisfaite au début de EBA. Cependant, notre condition de fusion est $e(\mathbf{x}) < e_t$. On résout ce problème en remplaçant la condition de descente par

$$e(\mathbf{x}+\Delta(\mathbf{\Delta}_2)) < e_t. \quad (13)$$

Un second problème est le suivant. D'un coté, l'approximation de Taylor dans l'éq. 11 requiert un petit $||c(\mathbf{x})||$. De l'autre coté, $||c(\mathbf{x}^*)||$ est la distance entre des positions de SfM et GPS au début de EBA, qui peut être grande. On résout ce problème en re-définissant c pendant les itérations de EBA de sorte que l'éq. 13 est satisfaite.

Itération réussie Premièrement, on voit que l'éq. 13 est satisfaite si $||c(\mathbf{x})||$ et $||\mathbf{\Delta}_2||$ sont suffisamment petits. En effet, lorsque $c(\mathbf{x})$ et $\mathbf{\Delta}_2$ tendent vers 0, $\Delta(\mathbf{\Delta}_2)$ tend vers $\mathbf{0}$ (d'après l'éq. 10) et donc $e(\mathbf{x}+\Delta(\mathbf{\Delta}_2))$ tend vers $e(\mathbf{x})$ qui est inférieur strict à e_t (éq. 1).
Deuxièmement, on propose de remplacer c dans tous les calculs par

$$c_\alpha(\mathbf{x}) = c(\mathbf{x}) - \alpha c(\mathbf{x}^*) \text{ avec } \alpha \in [0,1]. \quad (14)$$

On remarque que $c_1(\mathbf{x}^*) = \mathbf{0}$ et $c_0(\mathbf{x}) = c(\mathbf{x})$. On fait décroitre α progressivement durant les itérations de EBA

de 1 (aucune contrainte au début) à 0 (contrainte complète). La valeur finale de α peut être différente de 0 et cette valeur mesure le succès de la fusion entre le GPS et la vision de $\alpha = 1$ (échec) à $\alpha = 0$ (fusion réussie à 100%).
Troisièmement, on montre qu'une itération réussie est possible en faisant décroitre α. Supposons que $c_\alpha(\mathbf{x}) = \mathbf{0}$ avant l'itération courante (ceci est vrai avant toutes les itérations avec $\alpha = 1$). On a $c_{\alpha-\delta}(\mathbf{x}) = c_\alpha(\mathbf{x}) + \delta c(\mathbf{x}^*) = \delta c(\mathbf{x}^*)$. On peut donc faire décroitre α de sorte que $||c_\alpha(\mathbf{x})||$ est arbitrairement petit. Une fois le nouvel α choisi, c_α est re-défini et $\bar{\mathbf{g}}_2$ est calculé avec l'éq. 12. Nous pouvons choisir λ suffisamment grand tel que $||\mathbf{\Delta}_2||$ est arbitrairement petit grâce à $(\mathtt{H}_2 + \lambda \mathrm{diag}(\mathtt{H}_2))\mathbf{\Delta}_2 = -\bar{\mathbf{g}}_2$. Grâce à la discussion précédente, l'éq. 13 est satisfaite puisque $||c_\alpha(\mathbf{x})||$ et $||\mathbf{\Delta}_2||$ sont suffisamment petits. Donc l'itération est réussie et $\mathbf{x}$ est remplacé par $\mathbf{x} + \Delta(\mathbf{\Delta}_2)$. Maintenant le nouvel $\mathbf{x}$ annule le développement de Taylor à l'ordre 1 de c_α (éq. 9). Puisque c_α est linéaire dans notre cas, on a toujours $c_\alpha(\mathbf{x}) = \mathbf{0}$ et la méthode peut continuer.

4 Détails techniques

On explique maintenant comment implémenter efficacement IBA (partie 4.1), EBA (partie 4.2), et comment rendre la fusion plus facile en pratique (partie 4.3).

4.1 AF avec contrainte d'inégalité

Dans la partie 3.2, il faut résoudre efficacement $(\mathtt{H}_I + \lambda \mathrm{diag}(\mathtt{H}_I))\mathbf{\Delta} = -\mathbf{g}_I$. Soient $\tilde{\mathtt{H}}$ et $\tilde{\mathbf{g}}$ tels que

$$\mathtt{H}_I + \lambda \mathrm{diag}(\mathtt{H}_I) = \tilde{\mathtt{H}} + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T, \tilde{\mathbf{g}} = \sqrt{\frac{2\gamma}{(e_t - e)^3}}\mathbf{g}. \quad (15)$$

Un calcul simple permet de vérifier que

$$(\tilde{\mathtt{H}} + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T)^{-1} = (\mathtt{I} - \frac{\tilde{\mathtt{H}}^{-1}\tilde{\mathbf{g}}\tilde{\mathbf{g}}^T}{1 + \tilde{\mathbf{g}}^T\tilde{\mathtt{H}}^{-1}\tilde{\mathbf{g}}})\tilde{\mathtt{H}}^{-1}. \quad (16)$$

On introduit $\mathbf{a} = -\tilde{\mathtt{H}}^{-1}\mathbf{g}_I$, $\mathbf{b} = \tilde{\mathtt{H}}^{-1}\tilde{\mathbf{g}}$, et on obtient

$$\mathbf{\Delta} = -(\tilde{\mathtt{H}} + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T)^{-1}\mathbf{g}_I = \mathbf{a} - \frac{\tilde{\mathbf{g}}^T\mathbf{a}}{1 + \tilde{\mathbf{g}}^T\mathbf{b}}\mathbf{b}. \quad (17)$$

On explique maintenant comment calculer $\mathbf{a}$ et $\mathbf{b}$. D'après les éqs. 7 et 15, $\tilde{\mathtt{H}}$ à la même structure creuse que $\mathtt{J}^T\mathtt{J}$. Plus précisément [4], on a $\tilde{\mathtt{H}} = \begin{pmatrix} \mathtt{U} & \mathtt{W} \\ \mathtt{W}^T & \mathtt{V} \end{pmatrix}$ et $\tilde{\mathtt{H}} > 0$ avec $\mathtt{U}$ une matrice diagonale par blocs 6×6, $\mathtt{V}$ une matrice diagonale par blocs 3×3, et $\mathtt{W}$ une matrice de blocs 6×3 tels que le bloc (i, j) est nul si le j-ième point 3D n'est pas vu dans la i-ième image. Donc les systèmes linéaires $\tilde{\mathtt{H}}\mathbf{a} = -\mathbf{g}_I$ et $\tilde{\mathtt{H}}\mathbf{b} = \tilde{\mathbf{g}}$ sont résolus avec la même méthode creuse [4] et performante que celle utilisée pour le système linéaire de LM sans contrainte $(\mathtt{H} + \lambda \mathrm{diag}(\mathtt{H}))\mathbf{\Delta} = -\mathbf{g}$.

Algorithme IBA On rédige ici l'algorithme avec le style du langage C. Les entrées sont la fonction d'erreur de reprojection $e(\mathbf{x}) = ||E(\mathbf{x})||^2$, le(s) position(s) GPS $\mathbf{x}_1^{gps}$, la valeur initiale de $\mathbf{x}$ qui minimise e (i.e. $\mathbf{x} = \mathbf{x}^*$), un nombre maximum It_{max} d'itérations, et un seuil e_t tel que $e(\mathbf{x}) < e_t$. La sortie est $\mathbf{x}$ tel que $e(\mathbf{x}) < e_t$ et $e_I(\mathbf{x})$ est le plus petit possible.

$err = \gamma/(e_t - e(\mathbf{x})) + ||\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}||^2$; // $\mathtt{P} = \begin{pmatrix} \mathtt{I} & \mathtt{0} \end{pmatrix}$
$UpdateD = 1$; $\lambda = 0.001$;
for ($It = 0$; $It < It_{max}$; It++) {
  // mise a jour des derivees puis estimation de $\mathbf{\Delta}$
  if ($UpdateD$) {
    $UpdateD = 0$;
    $\mathbf{g} = 2\mathtt{J}^T E(\mathbf{x})$; $\mathtt{H} = 2\mathtt{J}^T\mathtt{J}$; // $\mathtt{J}$ est la jacobienne de E
    $\mathbf{g}_I = \frac{\gamma}{(e_t - e)^2}\mathbf{g} + 2\mathtt{P}^T(\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps})$; $\mathtt{H} = \frac{\gamma}{(e_t - e)^2}\mathtt{H} + 2\mathtt{P}^T\mathtt{P}$;
    $\tilde{\mathbf{g}} = \sqrt{\frac{2\gamma}{(e_t - e)^3}}\mathbf{g}$; // maintenant, $\mathtt{H}_I = \mathtt{H} + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T$
  }
  $\tilde{\mathtt{H}} = \mathtt{H} + \lambda \mathrm{diag}(\mathtt{H} + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T)$; // ne pas stocker $\mathtt{H}_I$
  resoudre $\tilde{\mathtt{H}}\begin{pmatrix} \mathbf{a} & \mathbf{b} \end{pmatrix} = \begin{pmatrix} -\mathbf{g}_I & \tilde{\mathbf{g}} \end{pmatrix}$
  $\mathbf{\Delta} = \mathbf{a} - (\tilde{\mathbf{g}}^T\mathbf{a})/(1 + \tilde{\mathbf{g}}^T\mathbf{b})\mathbf{b}$;
  // essayer de faire decroitre e_I
  if ($e(\mathbf{x} + \mathbf{\Delta}) \geq e_t$) { $\lambda = 10\lambda$; continue ; }
  $err' = \gamma/(e_t - e(\mathbf{x} + \mathbf{\Delta})) + ||\mathtt{P}(\mathbf{x} + \mathbf{\Delta}) - \mathbf{x}_1^{gps}||^2$;
  if ($err' < err$) {
    $\mathbf{x} = \mathbf{x} + \mathbf{\Delta}$;
    if ($0.9999err < err'$) break ; // convergence trop lente
    $err = err'; UpdateD = 1; \lambda = \lambda/10$;
  } else $\lambda = 10\lambda$;
}

4.2 AF avec contrainte d'égalité

Comment estimer efficacement $\mathbf{\Delta}_2$? Le calcul de $\mathbf{\Delta}_2$ dans une itération réussie de EBA s'écrit de la façon concise suivante : trouver δ et λ positifs stricts tels que

$$\begin{aligned} &\bar{\mathbf{g}}_2 = \mathbf{g}_2 - \mathtt{H}_{21}c_{\alpha-\delta}(\mathbf{x}) \\ &(\mathtt{H}_2 + \lambda \mathrm{diag}(\mathtt{H}_2))\mathbf{\Delta}_2 = -\bar{\mathbf{g}}_2 \\ &e(\mathbf{x} + \begin{pmatrix} -c_{\alpha-\delta}(\mathbf{x})^T & \mathbf{\Delta}_2^T \end{pmatrix}^T) < e_t. \end{aligned} \quad (18)$$

Le calcul le plus coûteux est celui du système linéaire. A première vue, il faut le résoudre pour chaque (λ, δ) que l'on essaye puisque $\bar{\mathbf{g}}_2$ dépend de $c_{\alpha-\delta}(\mathbf{x})$. Heureusement, on peut réduire le nombre de ces résolutions. Si on résout

$$(\mathtt{H}_2 + \lambda \mathrm{diag}(\mathtt{H}_2))\begin{pmatrix} \mathbf{\Delta}_2^a & \mathbf{\Delta}_2^b \end{pmatrix} = \begin{pmatrix} -\mathbf{g}_2 & \mathtt{H}_{21} \end{pmatrix}, \quad (19)$$

on obtient $\mathbf{\Delta}_2 = \mathbf{\Delta}_2^a + \mathbf{\Delta}_2^b c_{\alpha-\delta}(\mathbf{x})$. On voit maintenant comment améliorer : une fois que le système linéaire de l'éq. 19 est résolu pour une valeur de λ, $\mathbf{\Delta}_2$ s'obtient très rapidement pour tous les δ que l'on essaye.

Algorithme EBA On essaye $\delta \in \{\alpha, \alpha/2, \cdots \alpha/2^{10}\}$ dans l'ordre décroissant. Si tous ces δ échouent, on change l'itération d'EBA avec $\delta = 0$. Ceci implique que $\mathbf{\Delta}^T = \begin{pmatrix} \mathbf{0}^T & (\mathbf{\Delta}_2^a)^T \end{pmatrix}$. Ensuite on trouve λ tel que $e(\mathbf{x} + \mathbf{\Delta}) < e(\mathbf{x})$ comme dans le cas non contraint.
En pratique, on alterne une itération réussie avec $\delta > 0$ (itération "E") et une itération réussie avec $\delta = 0$ (itération "U") pour décroitre e dés que possible. Si $\alpha = 0$, seules les itérations "U" sont appliquées jusqu'à convergence.

L'algorithme suivant, rédigé avec le style du langage C, donne les détails manquants. Les entrées sont la fonction d'erreur de reprojection $e(\mathbf{x}) = ||E(\mathbf{x})||^2$, la fonction de contrainte c, la valeur initiale de $\mathbf{x}$ qui minimise e, le nombre maximum It_{max} d'itérations, et le seuil e_t tel que $e(\mathbf{x}) < e_t$. La sortie est $(\mathbf{x}, \alpha)$ tel que $c_\alpha(\mathbf{x}) = \mathbf{0}$, $e(\mathbf{x}) < e_t$ avec le plus petit α possible.

$err = e(\mathbf{x})$; $\mathbf{c}^* = c(\mathbf{x})$;
$UpdateD = 1$; $\lambda = 0.001$; $\alpha = 1$;
$\alpha_{old} = 1$; // utile pour alterner les iterations "E" et "U"
for ($It = 0$; $It < It_{max}$; It++) {
  // mise a jour des derivees puis estimation de $\mathbf{\Delta}_2^a$ et $\mathbf{\Delta}_2^b$
  if ($UpdateD$) {
    $UpdateD = 0$;
    $\mathbf{g} = \mathtt{J}^T E(\mathbf{x})$; $\mathtt{H} = \mathtt{J}^T\mathtt{J}$; // J est la jacobienne de E
    $\begin{pmatrix}\mathbf{g}_1\\ \mathbf{g}_2\end{pmatrix} = \mathbf{g}$; $\begin{pmatrix}\mathtt{H}_1 & \mathtt{H}_{12}\\ \mathtt{H}_{21} & \mathtt{H}_2\end{pmatrix} = \mathtt{H}$;
  }
  resoudre $(\mathtt{H}_2 + \lambda \mathrm{diag}(\mathtt{H}_2))\begin{pmatrix}\mathbf{\Delta}_2^a & \mathbf{\Delta}_2^b\end{pmatrix} = \begin{pmatrix}-\mathbf{g}_2 & \mathtt{H}_{21}\end{pmatrix}$
  // iteration E : essayer de decroitre α en bornant e
  if ($0 < \alpha$ && $\alpha_{old} == \alpha$) {
    for ($It_2 = 0, \alpha' = 0$; $It_2 < 10$; It_2++) {
      $c_{\alpha'}(\mathbf{x}) = c(\mathbf{x}) - \alpha'\mathbf{c}^*$; $\mathbf{\Delta}_2 = \mathbf{\Delta}_2^a + \mathbf{\Delta}_2^b c_{\alpha'}(\mathbf{x})$;
      $\mathbf{\Delta}^T = \begin{pmatrix}-c_{\alpha'}(\mathbf{x})^T & \mathbf{\Delta}_2^T\end{pmatrix}$; $err' = e(\mathbf{x} + \mathbf{\Delta})$;
      if ($err' < e_t$) break ; // succes si vrai
      $\alpha' = \frac{1}{2}(\alpha + \alpha')$;
    }
    if ($It_2 < 10$) { // succes si vrai
      $\alpha_{old} = \alpha; \alpha = \alpha'; \mathbf{x} = \mathbf{x} + \mathbf{\Delta}$;
      $err = err'; UpdateD = 1$; continue ;
    }
  }
  // iteration U : essayer de decroitre e avec α constant
  $\mathbf{\Delta}_2 = \mathbf{\Delta}_2^a$; $\mathbf{\Delta}^T = \begin{pmatrix}\mathbf{0}^T & \mathbf{\Delta}_2^T\end{pmatrix}$; $err' = e(\mathbf{x} + \mathbf{\Delta})$;
  if ($err' < err$) {
    $\mathbf{x} = \mathbf{x} + \mathbf{\Delta}$;
    if ($\alpha == 0$ && $0.9999 err < err'$) break ;
    $\alpha_{old} = \alpha, err = err'; UpdateD = 1; \lambda = \lambda/10$;
  } else $\lambda = 10\lambda$;
}

4.3 Faciliter la fusion en pratique

Premièrement, on montre un lien entre une covariance et l'accroissement de e due à la fusion GPS-vision. Avant la fusion, $\mathbf{x} = \mathbf{x}^*$ et $\mathbf{g} = \mathbf{0}$. Puis l'éq. 12 donne $\bar{e}_2 = e(\mathbf{x}) + 0.5c^T\mathtt{H}_1 c$ et $\bar{\mathbf{g}}_2 = -\mathtt{H}_{21}c$. De plus, l'approximation de Taylor (éq. 11) est minimisée si $\mathtt{H}_2\mathbf{\Delta}_2 = \mathtt{H}_{21}c$. Avec $\mathbf{\Delta}_2 = \mathtt{H}_2^{-1}\mathtt{H}_{21}c$, le minimum de l'éq. 11 est

$$\begin{aligned} e(\mathbf{x} + \Delta(\mathbf{\Delta}_2)) &\approx \bar{e}_2 - \mathbf{\Delta}_2^T\mathtt{H}_{21}c + .5\mathbf{\Delta}_2^T\mathtt{H}_2\mathbf{\Delta}_2 \\ &\approx e + .5c^T(\mathtt{H}_1 - \mathtt{H}_{12}\mathtt{H}_2^{-1}\mathtt{H}_{21})c \end{aligned} \quad (20)$$

Soit $\mathtt{C}_1$ la covariance de $\mathbf{x}_1$ dérivant de la mimisation de e : $\mathtt{C}_1$ est un bloc haut-gauche de $\mathtt{H}^{-1}$ multiplié par l'échelle du bruit image $\sigma^2 \in \mathbb{R}^+$. En utilisant le complément de Schur [13] de $\mathtt{H}_2$ dans $\mathtt{H}$, on a $\mathtt{C}_1 = \sigma^2(\mathtt{H}_1 - \mathtt{H}_{12}\mathtt{H}_2^{-1}\mathtt{H}_{21})^{-1}$.

On obtient le résultat suivant : si $\mathbf{x}$ minimise l'erreur de reprojection e et $\mathbf{\Delta}_1$ est la correction idéale de $\mathbf{x}_1$ pour fusionner (i.e. $\mathbf{\Delta}_1 = \mathbf{x}_1^{gps} - \mathbf{x}_1 = -c(\mathbf{x})$), la valeur minimale de $\mathbf{\Delta}_2 \mapsto e(\mathbf{x}_1 + \mathbf{\Delta}_1, \mathbf{x}_2 + \mathbf{\Delta}_2)$ est

$$e(\begin{pmatrix}\mathbf{x}_1 + \mathbf{\Delta}_1\\ \mathbf{x}_2 - \mathtt{H}_2^{-1}\mathtt{H}_{21}\mathbf{\Delta}_1\end{pmatrix}) \approx e(\mathbf{x}) + 0.5\sigma^2\mathbf{\Delta}_1^T\mathtt{C}_1^{-1}\mathbf{\Delta}_1. \quad (21)$$

Deuxièmement, on explique comment l'éq. 21 facilite la fusion. Dans nos expériences, les AFs de fusion sont des AFs locaux et $\mathbf{x}$ rassemble les paramètres 3D des k images clefs les plus récentes. Le repère de reconstruction est fixée au début et $\mathbf{x}_1$ est à la fin de ces k images. Dans ces conditions, $\mathtt{C}_1$ croit lorsque k croit. Grâce à l'éq. 21, plus $\mathtt{C}_1$ est grande, plus l'accroissement de e du à la correction $\mathbf{\Delta}_1$ est petit. Ceci signifie qu'un grand k facilite la fusion GPS-vision. Ce résultat est seulement valide pour un $\mathbf{\Delta}_1$ suffisamment petit car l'éq.21 vient d'une approximation de Taylor. Nos expériences confirment que k doit être suffisamment grand pour les AFs de fusion.
Enfin, la méthode suivante est aussi utilisée pour faciliter la fusion : on borne la longueur des suivis (tracks) de points dans les images (ex : un suivi sur 13 vues successives est remplacé par 2 suivis sur 5 vues et 1 suivi sur 3 vues).

5 Experimentations

5.1 Intégration de la fusion dans le SfM

La méthode de SfM [11] reconstruit le tout début de séquence avec des méthodes standard et ensuite alterne les étapes suivantes : (1) une nouvelle image clef est sélectionnée et des points d'intérêts sont appariés avec ceux de l'image clef précédente par corrélation (2) la nouvelle pose est estimée avec la méthode de Grunert et RANSAC (3) de nouveaux points 3D sont reconstruits à partir des nouveaux appariements et (4) un AFL raffine la géométrie des n images clefs les plus récentes. Dans le contexte de l'AFL, $\mathbf{x}$ rassemble les poses 6D des n images clefs les plus récentes et les points 3D qui ont au moins une observation dans ces images, $e(\mathbf{x})$ est la somme des carrés des erreurs de reprojection de ces points dans les N images clefs les plus récentes ($n = 3$, $N = 10$), et $\mathtt{H} > 0$.
Cet article ajoute l'étape (5), une étape de fusion qui est la version locale de UBA, IBA ou EBA. Plus précisement, le $e(\mathbf{x})$ impliqué dans les AFs de fusion est l'erreur de l'AFL qui raffine la géométrie des k images clefs les plus récentes. Le minimiseur $\mathbf{x}^*$ de e est estimé avant chaque AF de fusion grâce à une itération de cet AFL. Le vecteur $\mathbf{x}_1$ est la localisation 3D de l'image clef la plus récente. Notons que l'étape (5) n'implique pas d'outliers, car ceux ci sont rejetés dans l'étape (4) comme dans [11].
Le choix de k est important : un k petit est mieux pour le temps de calcul, mais un k grand rend la fusion plus facile (partie 4.3). La valeur de k sera donnée plus tard. Le seuil e_t contrôle le compromis entre la vision et le GPS dans le résultat de fusion. Plus e_t est petit, plus la vision s'impose dans la fusion, moins les erreurs de GPS sont tolérées. Si $e_t = e(\mathbf{x}^*)$, le résultat de la fusion est celui de

la vision ($\mathbf{x}^*$) et le GPS est ignoré. Dans cet article, on choisit $e_t = 1.05^2 e(\mathbf{x}^*)$. Autrement dit, un accroissement de reprojection d'erreur (RMS) de 5% est accepté pour la fusion. Les autres paramètres de UBA et IBA sont

$$\beta = \frac{e(\mathbf{x}^*)}{||\mathrm{P}\mathbf{x}^* - \mathbf{x}_1^{gps}||^2},\ \gamma = \frac{e_t - e(\mathbf{x}^*)}{10}||\mathrm{P}\mathbf{x}^* - \mathbf{x}_1^{gps}||^2. \quad (22)$$

Ces poids sont tels que le rapport entre les termes de vision et GPS dans e_U (respectivement e_I) est 1 (respectivement 10) avant l'optimisation de fusion.

L'étape (5) s'applique dans la boucle principale dès que le résultat de vision est recalé dans le repère du GPS. La méthode de recalage est la suivante. D'abord, on prend deux instants $t_0 = 0$ et t_1 tels que la distance entre les deux positions GPS dépasse $10m$. Ensuite on définit la direction verticale dans le résultat de vision en supposant que l'axe x et le mouvement de la caméra sont horizontaux entre t_0 et t_1. Maintenant, 3 points sont définis dans les deux systèmes de coordonnées (vision et GPS) et une fonction similitude est estimée à partir de ces points. On recale enfin la vision dans le repère GPS avec cette fonction.

5.2 Conditions expérimentales et notations

Le GPS et la caméra sont fixés sur une voiture. La trajectoire fait environ 4 km, comporte des lignes droites et des virages serrés, des ronds points, et des arrêts temporaires dus à des feux rouges. La scène inclue des immeubles, des arbres, et d'autres véhicules en déplacement.

Le GPS est bas coût et fournit une pose 2D (longitude, latitude) à une fréquence de 1 Hz. L'altitude est forcée à 0. Une fois que les coordonnées GPS sont converties en coordonnées euclidiennes 3D en mètres, une position GPS s'obtient pour tout instant par interpolation linéaire. La vérité terrain est fournie par un IXSEA LandINS (associé avec un GPS RTK qui lui n'est pas bas coût) à une fréquence de 10 Hz. Les moyennes, écart types et maxima de l'erreur du GPS bas coût dans notre séquence sont $4.28m$, $2.34m$, et $12.2m$, respectivement.

La caméra est monoculaire et calibrée, pointe vers l'avant et fournit des images 640×352 (fig. 1) à une fréquence de 25 Hz. 2480 images clefs sont sélectionnées parmi 14850 images, telles qu'il y a environ 600 points de Harris appariés entre deux images clefs consécutives. On suppose que la distance entre la caméra et le GPS est petite par rapport à l'erreur du GPS. Les coordonnées GPS de la caméra ($\mathbf{x}_1^{gps}$) sont donc approximées par celle de l'antenne GPS.

La localisation 3D des images clefs est donnée par 6 méthodes : SfM (GPS ignoré), GPS (SfM ignoré), GT (vérité terrain ou "ground truth"), UBA, IBA et EBA. UBA, IBA et EBA fusionnent sous les mêmes conditions : mêmes images clefs, mêmes appariements, même nombre maximal d'itérations ($It_{max} = 4$) et même k.

Soient a et b deux méthodes différentes que l'on veut comparer. Soient l_a^i et e_a^i la localisation 3D et l'erreur de reprojection (RMS) fournie par la méthode a pour la i-ième image clef. On étudie la distribution de $\forall i, ||l_a^i - l_b^i||$ avec



FIGURE 1 – Quelques images de la séquence vidéo.

f	m_f^{gps}	σ_f^{gps}	∞_f^{gps}	m_f^{gt}	σ_f^{gt}	∞_f^{gt}
SfM	165	172	591	163	172	592
UBA	2.62	2.39	11.2	5.58	3.18	14.0
IBA	1.23	1.50	8.46	4.57	2.83	12.1
EBA	2.48	2.27	10.5	5.49	3.11	14.0
GPS	0	0	0	4.28	2.34	12.2

TABLE 1 – Erreurs de localisation en mètres pour SfM,GPS, et les AFs de fusion (UBA, IBA et EBA avec $k = 40$).

$a \in \{\mathrm{SfM}, \mathrm{GPS}, \mathrm{UBA}, \mathrm{IBA}, \mathrm{EBA}\}$ et $b \in \{\mathrm{GPS}, \mathrm{GT}\}$. Ses moyennes, écart types et maxima sont m_a^b, σ_a^b et ∞_a^b. On étudie aussi la distribution de $\forall i, e_a^i / e_{\mathrm{SfM}}^i$ avec $a \in \{\mathrm{UBA}, \mathrm{IBA}, \mathrm{EBA}\}$. Ses moyennes, écart types et maxima sont m_a^{2d}, σ_a^{2d} et ∞_a^{2d}. On nomme dans la suite ces distributions par "erreurs de localisation" et "erreurs image". Elles sont estimées après la fusion complète de la séquence.

5.3 Comparaison de UBA, IBA et EBA

La table 1 montre les erreurs de localisation avec $k = 40$. Les trois fusions (UBA, IBA, EBA) réduisent beaucoup les erreurs de SfM par rapport au GPS (jusqu'à environ 2m). Les erreurs relative à la vérité terrain sont également grandement réduites jusqu'à environ 5m, ce qui est l'ordre de grandeur de l'erreur du GPS. Cependant, les méthodes de fusion ne sont pas capable d'améliorer la précision du GPS car les erreurs de fusion sont légèrement plus grandes que celle du GPS seul. D'après les valeurs de m_a^{GPS} et m_a^{GT}, les meilleurs résultats sont ceux de IBA, suivis par ceux de EBA et UBA. La fig. 4 montre les erreurs de localisation des trois fusions en fonction du numéro d'image clef.

La fig. 2 montre des vues de dessus globales des trajectoires GT, SfM et IBA, chacune étant superposée avec la trajectoire GPS. On distingue facilement les différences GPS-GT et GPS-SfM, et plus difficilement la différence GPS-IBA. La fig. 3 montre une vue locale des trajectoires fournie par les trois AFs de fusion, dans le cas où il y a de hauts bâtiments au bord de la route. La voiture se déplace de droite à gauche. A gauche, on voit que la forme

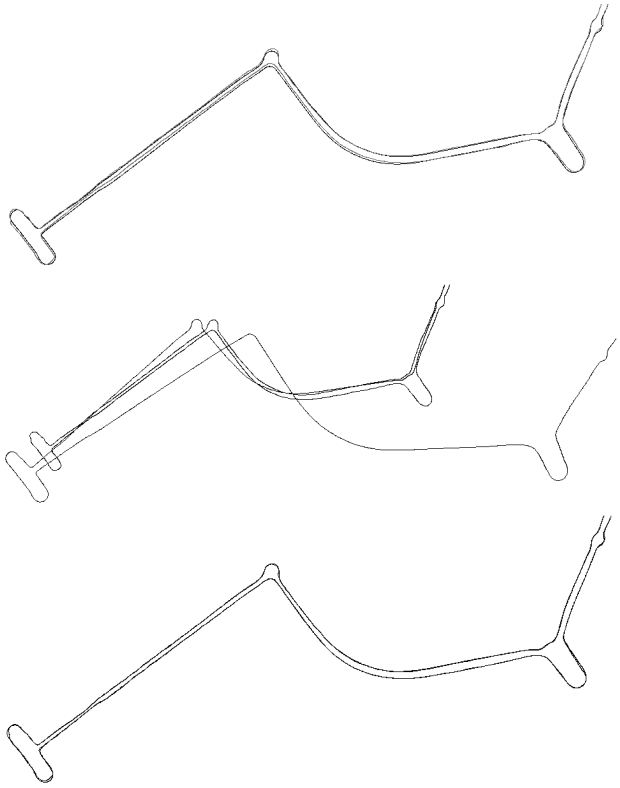


FIGURE 2 – De haut en bas : vues de dessus de GPS (gris) et GT (noir), de GPS et SfM, de GPS et IBA.

	k=40			k=30		
f	m_f^{2d}	σ_f^{2d}	∞_f^{2d}	m_f^{2d}	σ_f^{2d}	∞_f^{2d}
UBA	1.037	0.043	1.29	1.054	0.062	1.34
IBA	1.049	0.046	1.29	1.021	0.040	1.50
EBA	1.038	0.045	1.29	1.071	0.067	1.40

TABLE 2 – Erreurs images dues à la fusion.

des trajectoires des AFs de fusion sont meilleurs que celle du GPS : les trajectoires de fusion sont lisses comme celle de GT, celle du GPS (avec interpolation linéaire) n'est pas lisse en un point sur la gauche. On voit aussi que le GPS ne fournit pas localement un facteur d'échelle correct aux trajectoires des fusions.

Les erreurs images sont montrées à gauche dans la table 2 pour $k = 40$. On vérifie qu'elles sont acceptables pour tous les AFs de fusion car elles sont proches de 1.05, valeur utilisée pour définir le seuil e_t.

Les temps de calcul moyens de U-/I-/EBA sont respectivement 0.25, 0.27 et 0.28 secondes pour chaque image clef. Ces temps sont obtenus avec un PC portable core 2 duo 2.5Ghz, une implémentation creuse des hessiennes, et une factorisation de Cholesky du système de caméra réduit pour résoudre les systèmes linéaires de LM [13].

Les mêmes expériences sont faites avec $k = 30$. On voit

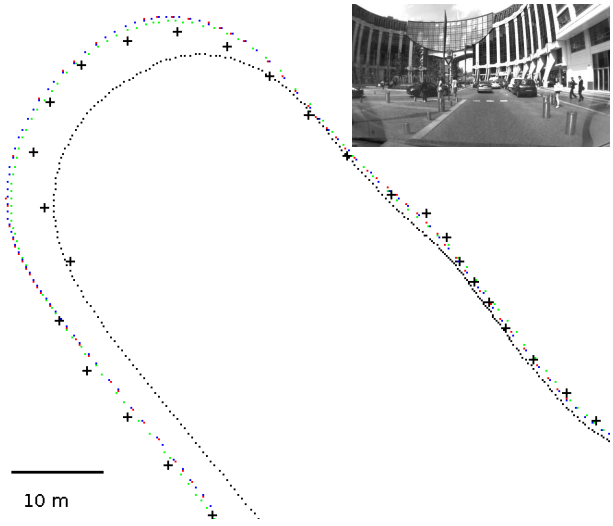


FIGURE 3 – Une vue locale de la localisation estimée par UBA (rouge), IBA (vert), EBA (bleu), GPS (croix noires), GT (points noirs). Des points rouges et bleus sont quasiment confondus en haut à gauche (1 point= 1 image clef).

f	m_f^{gps}	σ_f^{gps}	∞_f^{gps}	m_f^{gt}	σ_f^{gt}	∞_f^{gt}
UBA	22.7	38.5	195	24.4	37.3	193
IBA	43.2	47.1	250	43.3	46.7	248
EBA	1.80	1.7	8.3	4.99	3.09	13.1

TABLE 3 – Erreurs de localisation (en mètres) de la caméra pour les méthodes de fusion avec $k = 30$.

que la fusion est mauvaise pour UBA et IBA : les erreurs de localisation sont 10-40 fois plus grandes qu'avant (table 3). Cependant, leurs erreurs images sont acceptables (à droite dans la table 2). Le résultat de fusion est toujours correct pour EBA : l'erreur de localisation (relativement au GPS) est d'environ 1.8m et l'erreur image est légèrement plus grand (environ 1.07). D'autres expériences montrent que tous les AFs de fusion ont des erreurs de localisation trop grandes avec $k = 25$.

5.4 Choix de pondération pour UBA et IBA

Il faut choisir des poids β and γ pour définir UBA et IBA. C'est pourquoi on examine dans cette partie les effets de changements de ces poids autour de leurs valeurs par défaut données dans l'éq. 22. Les résultats sont donnés dans la table 4. On voit que les résultats de fusion restent acceptables si on multiplie ou divise les poids par 2. On voit aussi qu'un grand changement de poids (division ou multiplication par 10) produit de mauvais résultats de fusion. Ces expériences suggèrent que le réglage du poids est important, quoique pas trop difficile à trouver. De plus, elles confirment que IBA donne les meilleurs résultats de localisation.

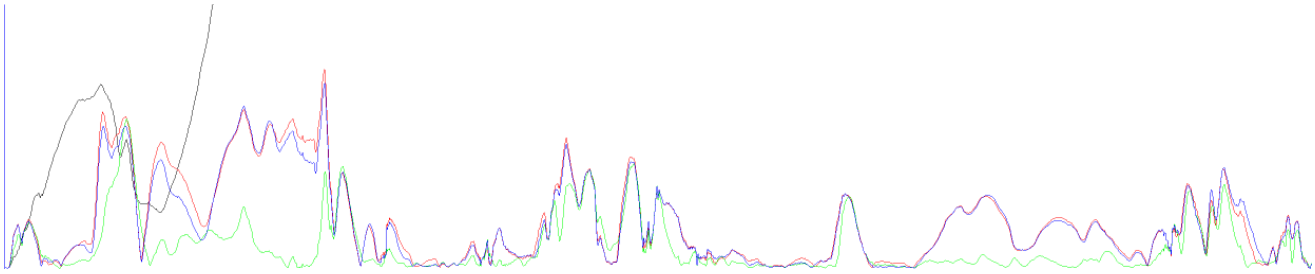


FIGURE 4 – Erreurs de localisation (par rapport au GPS) de SfM (noir), UBA (rouge), IBA (vert) and EBA (bleu) avec $k = 40$. L'abcisse donne le numéro d'image clef dans [0,2479] et l'ordonnée est l'erreur de localisation dans [0,15m].

f	w.	m_f^{gps}	∞_f^{gps}	m_f^{gt}	∞_f^{gt}	m_f^{2d}	∞_f^{2d}
UBA	$\frac{\beta}{10}$	135	409	133	409	1.00	1.11
UBA	$\frac{\beta}{2}$	2.66	11.3	5.62	14.0	1.04	1.31
UBA	2β	2.55	10.8	5.55	14.9	1.04	1.29
UBA	10β	405	1.3k	405	1.3k	1.02	1.19
IBA	$\frac{\gamma}{10}$	22.4	80.7	22.9	80.9	1.06	1.43
IBA	$\frac{\gamma}{2}$	1.88	9.84	4.88	12.3	1.06	1.32
IBA	2γ	1.64	12.5	4.78	12.1	1.05	1.26
IBA	10γ	195	690	193	691	1.00	1.10

TABLE 4 – Erreurs de localisation et image avec $k = 40$.

6 Conclusion

Deux AFs contraints IBA et EBA ont été introduits pour fusionner GPS et SfM. Le premier est contraint par une inégalité grâce à une fonction de pénalité, le second implique une contrainte d'égalité et est dérivé d'une méthode de réduction. Les algorithmes sont décrits en détails. On explique aussi pourquoi la difficulté de fusion augmente lorsque le nombre k de poses optimisées par les AFs de fusion décroit. Les expériences comparent les deux AFs avec la méthode existante UBA (qui minimise une somme pondérée d'erreurs images et GPS) dans le contexte du SfM incrémental appliqué sur une longue séquence urbaine.

Les trois AFs de fusion améliorent notablement les poses du SfM ; l'accroissement des erreurs de reprojection qui en résulte est faible. Grâce à la vérité terrain, on sait que la précision des poses obtenues est similaire à celle du GPS seul. La précision moyenne du GPS est cependant légèrement meilleure (c'est le seul capteur qui fournit des données absolues, la caméra monoculaire ne peut pas le faire). IBA produit les meilleurs résultats de fusion, EBA est classé second mais à deux avantages : c'est la méthode la moins sensible à un faible k (plus k est petit, plus rapide est le calcul) et elle ne dépend pas du choix d'un poids.

Des travaux futurs incluent des expériences avec d'autres AFs, l'amélioration du recalage initial de SfM dans le repère GPS, la comparaison des AFs dans un contexte non-incrémental, la fusion avec un GPS donnant l'altitude ou un autre capteur, la modélisation 3D géoréférencée.

Références

[1] Y. Bard, *Non-linear parameter estimation*, Academic Press Inc. (London) LTD, 1974.

[2] C. Ellum, *Integration of raw gps measurements into a bundle adjustment*, IAPRS series vol. XXXV, 2006.

[3] A. Eudes and M. Lhuillier, *Error propagation for local bundle adjustment*, CVPR'09.

[4] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2000.

[5] Y. Jeong, D. Nister, D. Steedly, R. Szeliski and I.S.Kweon, *Pushing the envelope of modern methods for bundle adjustment*, CVPR'10.

[6] H. Kume and T. Takemoti and T. Sato and N. Yokoya, *Extrinsic camera parameter estimation using video images and gps considering gps position accuracy*, ICPR'10.

[7] M. Lhuillier, *Fusion of gps and structure-from-motion using constrained bundle adjustments*, CVPR'11.

[8] P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome, *Towards geographical referencing of monocular slam reconstruction using 3d city models : application to real-time accurate vision-based localization*, CVPR'09.

[9] JC. McGlone, *The manual of photogrammetry, fifth edition*, ISBN-10 1570830711, ASPRS, 2004.

[10] J. Michot, A. Bartoli and F. Gaspard, *Bi-objective bundle adjustment with application to multi-sensor slam*, 3DPVT'10.

[11] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser and P. Sayd, *Generic and real-time structure-from-motion using local bundle adjustment*, Image and Vision Computing, 27 :1178-1193, 2009.

[12] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical recipies in c, second edition*, Cambridge University Press, 1999.

[13] B. Triggs, P. F. McLauchlan, R. Hartley and A. W. Fitzgibbon, *Bundle adjustment – a modern synthesis*, Vision Algorithms : Theory and Practice, 2000.