Fusion of GPS and Structure-from-Motion using Constrained Bundle Adjustments

Maxime Lhuillier

LASMEA-UMR 6602 Université Blaise Pascal/CNRS

63177 Aubière Cedex, France.

http://maxime.lhuillier.free.fr

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.

1. Introduction

Bundle adjustment (BA) is an iterative method of estimating camera poses and 3D points detected in an image sequence. The resulting poses and points minimize a sum of squared reprojection errors. Although BA has been known about for some time [14], it is still a field of research.

Recent developments mainly concern accelerations for long sequences. In [12], the optimization of the whole sequence is accelerated by optimizations of several sub-maps in parallel with their own local coordinate systems. Conjugated gradient may be faster than variable ordering and factorization to solve the reduced camera system [6]. Local BA [11] (LBA) is applied at several times to refine all 3D parameters of the most recent keyframes of a video. The resulting accuracy and uncertainty of the poses are similar to those of the global BA on the whole sequence [11, 4].

Another BA topic is fusion of data coming from several sensors. Fusion is useful for reducing the error accumulation of Structure-from-Motion (SfM), which is unavoidable for long image sequence (especially if the camera is monocular). Global BA is used in aerial Photogrammetry to combine image, inertial and GPS measures: the cost function minimized by BA is a sum of image, inertial and GPS errors weighted by measure covariances [9]. There is also an attempt to include the GPS pseudo-ranges directly as measures in BA [3]. In a different context, the reprojection errors of 3D points involved in BA are modified such that points are constrained into vertical planes stored in a GIS database [8]. Recent work combines GPS and image measures [7] (or inertial and image measures [10]) using LBA, which minimizes a weighted sum of GPS (or inertial) and image errors. In [7], the experiments are limited to a small sequence (70 m) and the GPS error is defined by a high order polynomial.

The alternative of LBA for real-time fusion of data coming from several sensors is the Kalman Filter (KF) and its extensions. The SLAM community in Robotics is very active with regard to this topic [1]. KFs are more subject to prior knowledge of state covariance than LBA. Furthermore, it is recognized that KFs are less accurate than LBA (there is a comparison for image-inertial fusion in [10]).

Our paper has several contributions. Firstly, Section 2 provides a brief overview of six BAs which may solve the SfM-GPS fusion problem. Only sparse Levenberg-Marquardt [14] or LM (second order) methods are considered here for efficiency. Secondly, Section 3 introduces our two constrained BA for fusion. The first one involves inequality constraint and the second one involves equality constraint. These BAs enforce an upper bound for the reprojection error, while the other fusion BAs [9, 10, 7] do not guarantee a small reprojection error and requires a weight. Thirdly, Section 4 provides important technical information (how to facilitate and accelerate our fusion BAs). Finally, Section 5 compares the results of the fusion BAs in a context which is useful for applications: the incremental SfM based on LBA [11]. In experiments, our low cost GPS and monocular (calibrated) camera are mounted on a car moving in urban area.

2. BA candidates for SfM-GPS fusion

2.1. Main notations and assumptions

The Euclidean norm is $||.||$. Different fonts are used for vectors (*e.g.* $\mathbf{x}$), matrices (*e.g.* $\mathtt{H}$) and function/real (*e.g.* e). Assume $\mathtt{H} > 0$, *i.e.* $\mathtt{H}$ is positive definite. Let $\mathtt{D}$ and diag($\mathtt{H}$) be diagonal block of $\mathtt{H}$ and the diagonal matrix obtained by forcing to 0 the off-diagonal coefficients of $\mathtt{H}$. Reminder of properties of $\mathtt{H}$: $\mathtt{D} > 0$, diag($\mathtt{H}$) > 0, $\mathtt{H} + \mathtt{A}^T\mathtt{A} > 0$, $\mathtt{H}^{-1} > 0$.

Vector $\mathbf{x}$ is the 3D parameters (camera poses and 3D points) and $e(\mathbf{x})$ is the sum of squares of reprojection errors of $\mathbf{x}$. In this paper, we assume that the starting/input $\mathbf{x}$ of the fusion BAs is the minimizer $\mathbf{x}^*$ of e ($\forall \mathbf{x}, e(\mathbf{x}^*) \leq e(\mathbf{x})$).

Let $\mathbf{x}_1$ be location(s) of the camera. The variable ordering is such that $\mathbf{x}^T = \begin{pmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T \end{pmatrix}$. Let $\mathbf{x}_1^{gps}$ be the location(s) of the camera provided by GPS at the same time(s). Assuming that the GPS drift is bounded and that of SfM is not, the ideal output $\mathbf{x}$ of the fusion BAs meet $\mathbf{x}_1 \approx \mathbf{x}_1^{gps}$. We use shortened notation $e(\mathbf{x}_1, \mathbf{x}_2) = e((\mathbf{x}_1^T \quad \mathbf{x}_2^T)^T)$.

Let e_t be a threshold which is slightly greater than the minimum $e(\mathbf{x}^*)$ of e. In our context, the final/output $\mathbf{x}$ is assumed to be acceptable if its reprojection error is similar to the minimum of e, *i.e.* $e(\mathbf{x}) < e_t$.

2.2. BAs without explicit constraint

These BAs [9] are used to combine measurements from different sensors. We refer to them as "unconstrained BAs".

A first and simple approach is the BA which minimizes $\mathbf{x}_2 \mapsto e(\mathbf{x}_1^{gps}, \mathbf{x}_2)$ after replacing the starting $\mathbf{x}_1$ by $\mathbf{x}_1^{gps}$. This BA may converge if the starting value of $\mathbf{x}_1 - \mathbf{x}_1^{gps}$ is small enough. However, this value may be large in our case.

A second approach is the minimization of a sum of weighted errors of the two different sensors: we minimize

$$e_U(\mathbf{x}) = e(\mathbf{x}) + \beta||\mathbf{x}_1 - \mathbf{x}_1^{gps}||^2. \tag{1}$$

Here the problems are the adequate choice of weight β and the risk of inlier loss due to the term $\beta||\mathbf{x}_1 - \mathbf{x}_1^{gps}||^2$. The inliers are the detected points involved in e such that the reprojection error is less than a threshold. These problems are similar if we generalize $\beta||.||^2$ by a quadratic form defined by a covariance matrix.

2.3. BAs with equality constraint

Two methods are possible [14]: sequential quadratic programming (SQP) and reduction method. Both deal with the minimization of $e(\mathbf{x})$ subject to constraint $c_E(\mathbf{x}) = 0$. One iteration of these constrained BAs improves $\mathbf{x}$ by adding step $\mathbf{\Delta}$ subject to the linearized constraint $c_E(\mathbf{x} + \mathbf{\Delta}) \approx c_E(\mathbf{x}) + \frac{\partial c_E}{\partial \mathbf{x}}\mathbf{\Delta} = \mathbf{0}$. Like unconstrained BA, damping is used to define $\mathbf{\Delta}$ between the Gauss-Newton step, which minimizes the quadratic Taylor expansion of e, and a gradient descent step. The Taylor expansions require a small

enough $\mathbf{\Delta}$, which in turn requires a small enough value of $c_E(\mathbf{x}) = -\frac{\partial c_E}{\partial \mathbf{x}}\mathbf{\Delta}$.

SQP simultaneously estimates $\mathbf{\Delta}$ and the Lagrange multipliers for c_E. The reduction method applies if $\frac{\partial c_E}{\partial \mathbf{x}_1}$ is square and invertible. In this case, step $\mathbf{\Delta}_1$ of $\mathbf{x}_1$ is expressed from step $\mathbf{\Delta}_2$ of $\mathbf{x}_2$, and the quadratic Taylor expansion of e only depends on $\mathbf{\Delta}_2$.

2.4. BAs with inequality constraint

Other methods exist [2]: projection method and penalty function. In our context, the iterations of these constrained BAs enforce the inequality constraint of Section 2.1, *i.e.* $c_I(\mathbf{x}) > 0$ where $c_I(\mathbf{x}) = e_t - e(\mathbf{x})$.

A first approach is the minimization of

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

where $\gamma > 0$. Here $\mathbf{x}_1 - \mathbf{x}_1^{gps}$ is minimized while the penalty function $\gamma/c_I(\mathbf{x})$ enforces the inequality constraint. Penalty is the main (positive infinite) term in the neighborhood of $c_I(\mathbf{x}) = 0$, and it does not change the minimizers too much of $\mathbf{x} \mapsto ||\mathbf{x}_1 - \mathbf{x}_1^{gps}||^2$ elsewhere.

The projection method minimizes function $h(\mathbf{x})$ subject to constraint $c_I(\mathbf{x}) \geq 0$. Here one iteration starts by calculation of step $\mathbf{\Delta}$ of the unconstrained BA minimizing h and ignoring c_I. Firstly, let's assume $c_I(\mathbf{x}) > 0$. If $c_I(\mathbf{x} + \mathbf{\Delta}) < 0$, $\mathbf{\Delta}$ is reset by $\gamma\mathbf{\Delta}$ where $0 < \gamma < 1$ and $c_I(\mathbf{x} + \gamma\mathbf{\Delta}) = 0$. Secondly, let's assume $c_I(\mathbf{x}) = 0$. If $c_I(\mathbf{x} + \mathbf{\Delta}) < 0$, $\mathbf{\Delta}$ is reset using iteration of constrained BA minimizing h subject to equality constraint $c_I(\mathbf{x}) = 0$. Now $\mathbf{\Delta}$ is added to $\mathbf{x}$ in all cases.

2.5. Our fusion BAs

UBA is the BA which minimizes e_U (Eq. 1). We introduce two fusion methods **IBA** and **EBA**. IBA is the BA with inequality constraint which minimizes e_I (Eq. 2). Although the principle is simple, such an IBA was not used before for fusion or SfM. EBA is a BA with equality constraint, which is derived from the reduction method (Section 2.3). The derivation is not so easy: on the one hand the original BA requires small $c_E(\mathbf{x})$, while on the other $c_E(\mathbf{x}) = \mathbf{x}_1 - \mathbf{x}_1^{gps}$ may be large (SQP has the same problem).

Remember that the three fusion methods have the following properties. The input $\mathbf{x}$ is the minimizer $\mathbf{x}^*$ of e. The output $\mathbf{x}$ has a small reprojection error, *i.e.* $e(\mathbf{x}) < e_t$ (EBA should be designed to meet this constraint; the UBA output is ignored if $e(\mathbf{x}) > e_t$). Last, the sub-vector $\mathbf{x}_1$ of the output $\mathbf{x}$ is as close as possible to $\mathbf{x}_1^{gps}$.

3. Iteration of BAs

Section 3 describes the iterations of LM, IBA and EBA (the former is useful to explain the latters). In all cases, we check that successful iteration is possible.

The quadratic Taylor expansion of e is

$$e(\mathbf{x} + \boldsymbol{\Delta}) \approx e(\mathbf{x}) + \mathbf{g}^T\boldsymbol{\Delta} + 0.5\boldsymbol{\Delta}^T\mathtt{H}\boldsymbol{\Delta} \qquad (3)$$

where $\mathbf{g}$ and $\mathtt{H}$ are the gradient and hessian. The projection function $E : \mathbb{R}^n \to \mathbb{R}^m$ meets $e(\mathbf{x}) = ||E(\mathbf{x})||^2$. Let $\mathtt{J}$ be the jacobian of E at $\mathbf{x}$. We have $\mathbf{g} = 2\mathtt{J}^T E(\mathbf{x})$, the Gauss-Newton approximation $\mathtt{H} \approx 2\mathtt{J}^T\mathtt{J}$, and assume $\mathtt{H} > 0$.

3.1. Levenberg-Marquardt without constraint

The LM iteration to minimize $e(\mathbf{x})$ without constraint is the following [13, 14] (UBA minimizes a different function using LM). Efficient sparse methods are used to solve $(\mathtt{H} + \lambda\mathrm{diag}(\mathtt{H}))\boldsymbol{\Delta} = -\mathbf{g}$ for the current value of $\mathbf{x}$ and a damping coefficient $\lambda > 0$. If $e(\mathbf{x} + \boldsymbol{\Delta}) < e(\mathbf{x})$, the iteration is successful: $\mathbf{x}$ is replaced by $\mathbf{x} + \boldsymbol{\Delta}$ and λ is replaced by $\lambda/10$. Otherwise, λ is replaced by 10λ. Now several remarks are of use for continuing the paper.

Firstly, we show that successful iteration is possible. A large λ implies $\boldsymbol{\Delta} \approx \boldsymbol{\Delta}_d$ where $\lambda\mathrm{diag}(\mathtt{H})\boldsymbol{\Delta}_d = -\mathbf{g}$. The larger λ, the smaller $\boldsymbol{\Delta}$, and the better Taylor approximation of e. Furthermore, $\mathtt{H} > 0$ implies $(\mathtt{H} + \lambda\mathrm{diag}(\mathtt{H}))^{-1} > 0$. Now the linear Taylor approximation and $\mathbf{g} \neq \mathbf{0}$ provide

$$e(\mathbf{x} + \boldsymbol{\Delta}) \approx e(\mathbf{x}) - \mathbf{g}^T(\mathtt{H} + \lambda\mathrm{diag}(\mathtt{H}))^{-1}\mathbf{g} < e(\mathbf{x}). \quad (4)$$

Secondly, we show that a small value of λ may accelerate the convergence of the descent step $\boldsymbol{\Delta}_d$. A small λ implies $\boldsymbol{\Delta} \approx \boldsymbol{\Delta}_{gn}$ where $\mathtt{H}\boldsymbol{\Delta}_{gn} = -\mathbf{g}$. Step $\boldsymbol{\Delta}_{gn}$ is the Gauss-Newton step, which minimizes the quadratic Taylor approximation of e. This step provides faster convergence than $\boldsymbol{\Delta}_d$ if $\mathbf{x}$ is close enough to the minimizer of e [14].

3.2. BA with inequality constraint (IBA)

The method is the same as in Section 3.1, except for the calculation of $\boldsymbol{\Delta}$. Let x_i be a coefficient of $\mathbf{x}$, $\mathtt{P}$ such that $\mathbf{x}_1 = \mathtt{P}\left(\mathbf{x}_1^T \quad \mathbf{x}_2^T\right)^T$ and $f(\mathbf{x}) = \gamma/(e_t - e(\mathbf{x}))$. We have

$$\frac{\partial f}{\partial x_i} = \frac{\gamma}{(e_t - e)^2}\frac{\partial e}{\partial x_i}$$
$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\gamma}{(e_t - e)^3}\left((e_t - e)\frac{\partial^2 e}{\partial x_i \partial x_j} + 2\frac{\partial e}{\partial x_i}\frac{\partial e}{\partial x_j}\right). \quad (5)$$

Then, we use the Gauss-Newton approximation ($\mathtt{H} \approx 2\mathtt{J}^T\mathtt{J}$) and obtain the gradient and hessian of e_I:

$$\mathbf{g}_I = \frac{\gamma}{(e_t - e)^2}\mathbf{g} + 2\mathtt{P}^T(\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps})$$
$$\mathtt{H}_I \approx \frac{2\gamma}{(e_t - e)^3}\left((e_t - e)\mathtt{J}^T\mathtt{J} + \mathbf{g}\mathbf{g}^T\right) + 2\mathtt{P}^T\mathtt{P}. \quad (6)$$

Now, the linear system $(\mathtt{H}_I + \lambda\mathrm{diag}(\mathtt{H}_I))\boldsymbol{\Delta} = -\mathbf{g}_I$ is solved. This can not be solved as in the unconstrained case since $\mathtt{H}_I$ is not sparse due to the dense term $\mathbf{g}\mathbf{g}^T$. Section 4.1 provides an efficient method to solve this linear system.

Finally, we check that the e_I decrease is possible. Eq. 6, $e(\mathbf{x}) < e_t$ and $\mathtt{J}^T\mathtt{J} > 0$ imply $\mathtt{H}_I > 0$. Then the e_I decrease is proved as the e decrease in Eq. 4.

3.3. Original BA with equality constraint

Now the LM iteration to minimize $e(\mathbf{x})$ subject to constraint $c(\mathbf{x}) = 0$ is described [14]. We use block-wise notations

$$\begin{pmatrix}\mathbf{x}_1 & \boldsymbol{\Delta}_1 & \mathbf{g}_1 \\ \mathbf{x}_2 & \boldsymbol{\Delta}_2 & \mathbf{g}_2\end{pmatrix} = \begin{pmatrix}\mathbf{x} & \boldsymbol{\Delta} & \mathbf{g}\end{pmatrix}, \begin{pmatrix}\mathtt{H}_1 & \mathtt{H}_{12} \\ \mathtt{H}_{21} & \mathtt{H}_2\end{pmatrix} = \mathtt{H} \quad (7)$$

and jacobian $\begin{pmatrix}\mathtt{C}_1 & \mathtt{C}_2\end{pmatrix}$ of c at $\mathbf{x}$. In our case, $\mathtt{C}_1 = \mathtt{I}$, $\mathtt{C}_2 = \mathtt{0}$ and step $\boldsymbol{\Delta}$ is such that

$$c(\mathbf{x} + \boldsymbol{\Delta}) \approx c(\mathbf{x}) + \mathtt{C}_1\boldsymbol{\Delta}_1 + \mathtt{C}_2\boldsymbol{\Delta}_2 = c(\mathbf{x}) + \boldsymbol{\Delta}_1 = \mathbf{0}. \quad (8)$$

Then, $\boldsymbol{\Delta}$ is a function of $\boldsymbol{\Delta}_2$:

$$\Delta(\boldsymbol{\Delta}_2) = \left(-c(\mathbf{x})^T \quad \boldsymbol{\Delta}_2^T\right)^T. \quad (9)$$

Thanks to Eq. 3 and $\boldsymbol{\Delta} = \Delta(\boldsymbol{\Delta}_2)$, we obtain

$$e(\mathbf{x} + \Delta(\boldsymbol{\Delta}_2)) \approx \bar{e}_2 + \boldsymbol{\Delta}_2^T\bar{\mathbf{g}}_2 + 0.5\boldsymbol{\Delta}_2^T\mathtt{H}_2\boldsymbol{\Delta}_2 \quad (10)$$

where

$$\begin{aligned}\bar{e}_2 &= e(\mathbf{x}) - \mathbf{g}_1^T c(\mathbf{x}) + 0.5c(\mathbf{x})^T\mathtt{H}_1 c(\mathbf{x}) \\ \bar{\mathbf{g}}_2 &= \mathbf{g}_2 - \mathtt{H}_{21}c(\mathbf{x})\end{aligned} \quad (11)$$

Step $\boldsymbol{\Delta}_2$ meets $(\mathtt{H}_2 + \lambda\mathrm{diag}(\mathtt{H}_2))\boldsymbol{\Delta}_2 = -\bar{\mathbf{g}}_2$. Now the iteration is the same as in Section 3.1 using $\boldsymbol{\Delta} = \Delta(\boldsymbol{\Delta}_2)$.

3.4. Modified BA with equality constraint (EBA)

Against the Original BA   Assume that EBA is the original BA using $c(\mathbf{x}) = \mathbf{x}_1 - \mathbf{x}_1^{gps}$ (now we use $c = c_E$). A first problem is the descending condition $e(\mathbf{x} + \Delta(\boldsymbol{\Delta}_2)) < e(\mathbf{x})$ to test step $\boldsymbol{\Delta} = \Delta(\boldsymbol{\Delta}_2)$. In our fusion context, the initial value of $\mathbf{x}$ is $\mathbf{x}^*$, which minimizes e. So the descending condition can not be meet at the beginning of EBA. However, we remind that our condition for fusion is $e(\mathbf{x}) < e_t$. We solve this problem, substituting the descending condition by

$$e(\mathbf{x} + \Delta(\boldsymbol{\Delta}_2)) < e_t. \quad (12)$$

A second problem is the following. On the one hand, the Taylor approximation in Eq. 10 requires a small $||c(\mathbf{x})||$. On the other hand, $||c(\mathbf{x}^*)||$ is the difference between SfM location and GPS location at the beginning of EBA, which may be large. We solve this problem, resetting c during EBA iterations such that Eq. 12 is meet.

Can We Reset c ?   First we show that Eq. 12 is meet if both $||c(\mathbf{x})||$ and $||\boldsymbol{\Delta}_2||$ are small enough. If $||c(\mathbf{x})||$ and $||\boldsymbol{\Delta}_2||$ are small enough,

$$\begin{aligned}e(\mathbf{x} + \Delta(\boldsymbol{\Delta}_2)) &\approx \bar{e}_2 + \boldsymbol{\Delta}_2^T\bar{\mathbf{g}}_2 \\ &\approx \bar{e}_2 - \bar{\mathbf{g}}_2^T(\mathtt{H}_2 + \lambda\mathrm{diag}(\mathtt{H}_2))^{-1}\bar{\mathbf{g}}_2 \\ &< \bar{e}_2.\end{aligned} \quad (13)$$

The last inequality is given by $\mathrm{H}_2 > 0$. Then, a small enough $||c(\mathbf{x})||$ provides $\bar{e}_2 \leq e_t$ according to the $\bar{e}_2$ definition in Eq. 11 and $e(\mathbf{x}) < e_t$. Now we have Eq. 12.

Second, we propose to replace c in all calculations by

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

Note that $c_1(\mathbf{x}^*) = \mathbf{0}$ and $c_0(\mathbf{x}) = c(\mathbf{x})$. We decrease α progressively during EBA iterations from 1 (no constraint before all iterations) to 0 (full constraint). The final value of α may be different to 0 and this measures the success of fusion between GPS and image data from $\alpha = 1$ (failure) to $\alpha = 0$ (100% success).

Third, we show that successful iterations are possible by decreasing α. Let us assume that $c_\alpha(\mathbf{x}) = \mathbf{0}$ before the current iteration (this is true before all iterations using $\alpha = 1$). We have $c_{\alpha-\delta}(\mathbf{x}) = c_\alpha(\mathbf{x}) + \delta c(\mathbf{x}^*) = \delta c(\mathbf{x}^*)$. So we can decrease α such that $||c_\alpha(\mathbf{x})||$ is arbitrarily small. Once the new α is chosen, c_α is reset and $\bar{\mathbf{g}}_2$ is computed using Eq. 11. We can choose a large enough λ such that $||\boldsymbol{\Delta}_2||$ is arbitrarily small thanks to $(\mathrm{H}_2 + \lambda\mathrm{diag}(\mathrm{H}_2))\boldsymbol{\Delta}_2 = -\bar{\mathbf{g}}_2$. Thanks to the previous discussion, Eq. 12 is meet since both $||c_\alpha(\mathbf{x})||$ and $||\boldsymbol{\Delta}_2||$ are small enough. Thus the iteration is successful and $\mathbf{x}$ is replaced by $\mathbf{x} + \Delta(\boldsymbol{\Delta}_2)$. Now the new $\mathbf{x}$ nullifies the linear Taylor expansion of c_α (Eq. 8). Since c_α is linear in our case, we still have $c_\alpha(\mathbf{x}) = \mathbf{0}$ and the method can continue.

4. Implementation

Now we will explain how to implement efficiently IBA (Section 4.1), EBA (Section 4.2), and how to make fusion easier (Section 4.3).

4.1. BA with inequality constraint

In Section 3.2, $(\mathrm{H}_I + \lambda\mathrm{diag}(\mathrm{H}_I))\boldsymbol{\Delta} = -\mathbf{g}_I$ should be solved efficiently. Let $\tilde{\mathrm{H}}$ and $\tilde{\mathbf{g}}$ be such that

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

Basic computation shows that

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

We introduce $\mathbf{a} = -\tilde{\mathrm{H}}^{-1}\mathbf{g}_I$, $\mathbf{b} = \tilde{\mathrm{H}}^{-1}\tilde{\mathbf{g}}$, and obtain

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

Now we explain how to estimate $\mathbf{a}$ and $\mathbf{b}$. According to Eqs. 6 and 15, $\tilde{\mathrm{H}}$ has the sparse structure of $\mathrm{H} = 2\mathrm{J}^T\mathrm{J}$. More precisely [14], we have $\tilde{\mathrm{H}} = \begin{pmatrix} \mathrm{U} & \mathrm{W} \\ \mathrm{W}^T & \mathrm{V} \end{pmatrix}$ and $\tilde{\mathrm{H}} > 0$.

where U is a 6×6 block-wise diagonal matrix, V is a 3×3 block-wise invertible diagonal matrix, and W is a 6×3 block-wise matrix such that the (i,j) block is zero if the j-th 3D point is not seen in the i-th image. So linear systems $\tilde{\mathrm{H}}\mathbf{a} = -\mathbf{g}_I$ and $\tilde{\mathrm{H}}\mathbf{b} = \tilde{\mathbf{g}}$ are solved using the same efficient sparse method [5] as the LM linear system $(\mathrm{H}+\lambda\mathrm{diag}(\mathrm{H}))\boldsymbol{\Delta} = -\mathbf{g}$.

The algorithm in C style is the following. The inputs are reprojection error $e(\mathbf{x}) = ||E(\mathbf{x})||^2$, GPS location(s) $\mathbf{x}_1^{gps}$, initial $\mathbf{x}$ which minimizes e (*i.e.* $\mathbf{x} = \mathbf{x}^*$), maximum number of iterations It_{max}, and threshold e_t such that $e(\mathbf{x}) < e_t$. The output is $\mathbf{x}$ such that $e(\mathbf{x}) < e_t$ and $e_I(\mathbf{x})$ has the smallest possible value.

$err = \gamma/(e_t - e(\mathbf{x})) + ||\mathrm{P}\mathbf{x} - \mathbf{x}_1^{gps}||^2;$ // $\mathrm{P} = \begin{pmatrix} \mathrm{I} & 0 \end{pmatrix}$
$UpdateD = 1; \lambda = 0.001;$
for $(It = 0; It < It_{max}; It$++$)$ {
  // derivative update and estimation of $\boldsymbol{\Delta}$
  if $(UpdateD)$ {
    $UpdateD = 0;$
    $\mathbf{g} = 2\mathrm{J}^T E(\mathbf{x}); \mathrm{H} = 2\mathrm{J}^T\mathrm{J};$ // J is the jacobian of E at $\mathbf{x}$
    $\mathbf{g}_I = \frac{\gamma}{(e_t-e)^2}\mathbf{g} + 2\mathrm{P}^T(\mathrm{P}\mathbf{x} - \mathbf{x}_1^{gps}); \mathrm{H} = \frac{\gamma}{(e_t-e)^2}\mathrm{H} + 2\mathrm{P}^T\mathrm{P};$
    $\tilde{\mathbf{g}} = \sqrt{\frac{2\gamma}{(e_t-e)^3}}\mathbf{g};$ // now, $\mathrm{H}_I = \mathrm{H} + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T$ (don't store H_I)
  }
  $\tilde{\mathrm{H}} = \mathrm{H} + \lambda\mathrm{diag}(\mathrm{H} + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T);$
  solve $\tilde{\mathrm{H}}\begin{pmatrix} \mathbf{a} & \mathbf{b} \end{pmatrix} = \begin{pmatrix} -\mathbf{g}_I & \tilde{\mathbf{g}} \end{pmatrix}$
  $\boldsymbol{\Delta} = \mathbf{a} - (\tilde{\mathbf{g}}^T\mathbf{a})/(1 + \tilde{\mathbf{g}}^T\mathbf{b})\mathbf{b};$
  // try to decrease e_I
  if $(e(\mathbf{x} + \boldsymbol{\Delta}) \geq e_t)$ { $\lambda = 10\lambda;$ continue; }
  $err' = \gamma/(e_t - e(\mathbf{x} + \boldsymbol{\Delta})) + ||\mathrm{P}(\mathbf{x} + \boldsymbol{\Delta}) - \mathbf{x}_1^{gps}||^2;$
  if $(err' < err)$ {
    $\mathbf{x} = \mathbf{x} + \boldsymbol{\Delta};$
    if $(0.9999err < err')$ break; // convergence is too slow
    $err = err'; UpdateD = 1; \lambda = \lambda/10;$
  } else $\lambda = 10\lambda;$
}

4.2. Modified BA with equality constraint

How to efficiently estimate $\boldsymbol{\Delta}_2$? As a reminder, the calculation of $\boldsymbol{\Delta}_2$ in a successful EBA iteration is concisely written as: find positive δ and λ such that

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

Solving the linear system is the main calculation. At first glance, this should be done for each tried (λ, δ) since $\bar{\mathbf{g}}_2$ depends on $c_{\alpha-\delta}(\mathbf{x})$. Fortunately, we can reduce the number of these calculations. We solve $\boldsymbol{\Delta}_2^a$ and $\boldsymbol{\Delta}_2^b$ such that

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

and obtain $\boldsymbol{\Delta}_2 = \boldsymbol{\Delta}_2^a + \boldsymbol{\Delta}_2^b c_{\alpha-\delta}(\mathbf{x})$. Now we see the improvement: once the linear system in Eq. 19 is solved, $\boldsymbol{\Delta}_2$ is obtained very efficiently for all tried δ.

EBA algorithm We try $\delta \in \{\alpha, \alpha/2, \cdots \alpha/2^{10}\}$ in the decreasing order. If all δ above fail, we change the EBA iteration using $\delta = 0$. This implies that $\boldsymbol{\Delta}^T = \begin{pmatrix} \mathbf{0}^T & (\boldsymbol{\Delta}_2^a)^T \end{pmatrix}$. Then we find λ such that $e(\mathbf{x} + \boldsymbol{\Delta}) < e(\mathbf{x})$ as in unconstrained BA.

In practice, we alternate successful iteration with $\delta > 0$ (E-iteration) and successful iteration with $\delta = 0$ (U-iteration) to decrease e as much as possible. If $\alpha = 0$, only U-iterations are applied until convergence.

The following algorithm in C style provides the remaining details. The inputs are reprojection error $e(\mathbf{x}) = ||E(\mathbf{x})||^2$, constraint c, initial $\mathbf{x}$ which minimizes e, maximum number of iterations It_{max}, and threshold e_t such that $e(\mathbf{x}) < e_t$. The output is $(\mathbf{x}, \alpha)$ such that $c_\alpha(\mathbf{x}) = \mathbf{0}$, $e(\mathbf{x}) < e_t$ and the smallest α as possible.

$err = e(\mathbf{x}); \mathbf{c}^* = c(\mathbf{x});$
$UpdateD = 1; \lambda = 0.001; \alpha = 1;$
$\alpha_{old} = 1; //\ \alpha_{old}$ is used to alternate E- and U-iterations
for $(It = 0; It < It_{max}; It{+}{+})$ {
  // derivative update and estimation of $\boldsymbol{\Delta}_2^a$ and $\boldsymbol{\Delta}_2^b$
  if $(UpdateD)$ {
    $UpdateD = 0;$
    $\mathbf{g} = \mathtt{J}^T E(\mathbf{x}); \mathtt{H} = \mathtt{J}^T \mathtt{J}; //\ \mathtt{J}$ is the jacobian of E at $\mathbf{x}$
    $\begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{pmatrix} = \mathbf{g}; \begin{pmatrix} \mathtt{H}_1 & \mathtt{H}_{12} \\ \mathtt{H}_{21} & \mathtt{H}_2 \end{pmatrix} = \mathtt{H};$
  }
  solve $(\mathtt{H}_2 + \lambda \mathrm{diag}(\mathtt{H}_2)) \begin{pmatrix} \boldsymbol{\Delta}_2^a & \boldsymbol{\Delta}_2^b \end{pmatrix} = \begin{pmatrix} -\mathbf{g}_2 & \mathtt{H}_{21} \end{pmatrix}$
  // E-iteration: try to decrease α with bounded e
  if $(0 < \alpha\ \&\&\ \alpha_{old} == \alpha)$ {
    for $(It_2 = 0, \alpha' = 0; It_2 < 10; It_2{+}{+})$ {
      $c_{\alpha'}(\mathbf{x}) = c(\mathbf{x}) - \alpha' \mathbf{c}^*; \boldsymbol{\Delta}_2 = \boldsymbol{\Delta}_2^a + \boldsymbol{\Delta}_2^b c_{\alpha'}(\mathbf{x});$
      $\boldsymbol{\Delta}^T = \begin{pmatrix} -c_{\alpha'}(\mathbf{x})^T & \boldsymbol{\Delta}_2^T \end{pmatrix}; err' = e(\mathbf{x} + \boldsymbol{\Delta});$
      if $(err' < e_t)$ break; // success if true
      $\alpha' = \frac{1}{2}(\alpha + \alpha');$
    }
    if $(It_2 < 10)$ { // success if true
      $\alpha_{old} = \alpha; \alpha = \alpha'; \mathbf{x} = \mathbf{x} + \boldsymbol{\Delta};$
      $err = err'; UpdateD = 1;$ continue;
    }
  }
  // U-iteration: try to decrease e without decreasing α
  $\boldsymbol{\Delta}_2 = \boldsymbol{\Delta}_2^a; \boldsymbol{\Delta}^T = \begin{pmatrix} \mathbf{0}^T & \boldsymbol{\Delta}_2^T \end{pmatrix}; err' = e(\mathbf{x} + \boldsymbol{\Delta});$
  if $(err' < err)$ {
    $\mathbf{x} = \mathbf{x} + \boldsymbol{\Delta};$
    if $(\alpha == 0\ \&\&\ 0.9999 err < err')$ break;
    $\alpha_{old} = \alpha, err = err'; UpdateD = 1; \lambda = \lambda/10;$
  } else $\lambda = 10\lambda;$
}

4.3. Make fusion easier

Firstly, we will explain the link between covariance and the increase of e due to SfM-GPS fusion. Before fusion,

$\mathbf{x} = \mathbf{x}^*$ and $\mathbf{g} = \mathbf{0}$. Then Eq. 11 provides $\bar{e}_2 = e(\mathbf{x}) + 0.5 c^T \mathtt{H}_1 c$ and $\bar{\mathbf{g}}_2 = -\mathtt{H}_{21} c$. Furthermore, the quadratic Taylor expansion of e (Eq. 10) is minimized if $\mathtt{H}_2 \boldsymbol{\Delta}_2 = \mathtt{H}_{21} c$. Using $\boldsymbol{\Delta}_2 = \mathtt{H}_2^{-1} \mathtt{H}_{21} c$, the minimum of Eq. 10 is

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

Let $\mathtt{C}_1$ be the covariance of $\mathbf{x}_1$ derived from the minimization of e. So $\mathtt{C}_1$ is a top-left block of $\mathtt{H}^{-1}$ multiplied by image noise σ^2. Using the Schur complement [14] of $\mathtt{H}_2$ in $\mathtt{H}$, we have $\mathtt{C}_1 = \sigma^2 (\mathtt{H}_1 - \mathtt{H}_{12} \mathtt{H}_2^{-1} \mathtt{H}_{21})^{-1}$.

We obtain the following result: if $\mathbf{x}$ minimizes the reprojection error e and $\boldsymbol{\Delta}_1$ is the ideal correction of $\mathbf{x}_1$ by fusion (*i.e.* $\boldsymbol{\Delta}_1 = \mathbf{x}_1^{gps} - \mathbf{x}_1 = -c(\mathbf{x})$), the minimum value of $\boldsymbol{\Delta}_2 \mapsto e(\mathbf{x}_1 + \boldsymbol{\Delta}_1, \mathbf{x}_2 + \boldsymbol{\Delta}_2)$ is

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

Secondly, we will explain how Eq. 21 makes fusion easier. In our experiments, the fusion BAs are local BAs and $\mathbf{x}$ concatenates the 3D parameters of the k most recent keyframes. The gauge is fixed as the beginning and $\mathbf{x}_1$ is at the end of the k most recent keyframes. Under these conditions, $\mathtt{C}_1$ increases when k increases. Thanks to Eq. 21, the larger the covariance $\mathtt{C}_1$, the smaller the increase of e due to correction $\boldsymbol{\Delta}_1$. This means that a large k facilitates the SfM-GPS fusion. This result is only valid for small enough $\boldsymbol{\Delta}_1$ since Eq.21 is derived from a Taylor approximation (Eq.10). Our experiments confirm that the fusion BAs require a large enough k.

In practice, we also use an other way to make fusion easier: the track lengths of image matches are bounded (track whose length is larger than 5 is splitted in several tracks).

5. Experiments

5.1. Integrating fusion to LBA-based SfM

SfM [11] reconstructs the very beginning of the sequence using standard methods and then alternates the following steps: (1) a new keyframe is selected from the input video and interest points are matched with the previous keyframe using correlation (2) the new pose is estimated using Grunert's method and RANSAC (3) new 3D points are reconstructed from the new matches and (4) LBA refines the geometry of the n-most recent keyframes. In the LBA context, $\mathbf{x}$ concatenates the 6D poses of the n-most recent images and the 3D points which have observation(s) in these images, $e(\mathbf{x})$ is the sum of squared reprojection errors of these 3D points in the N most recent images. There is no gauge freedom and $\mathtt{H} > 0$. Step 4 uses $n = 3$ and $N = 10$.

Our paper adds step (5), a fusion step which is the local version of UBA, IBA or EBA. This means that the $e(\mathbf{x})$ involved in the fusion BAs is the reprojection error of the LBA

which refines the geometry of the k-most recent keyframes. The minimizer $\mathbf{x}^*$ of e is estimated before each fusion BA using a single iteration of this LBA (remind that LBA does not involve outliers). Vector $\mathbf{x}_1$ is the 3D location of the most recent key-frame.

Note that the choice of k is important: small k is better for fast computation, but large k makes fusion easier (Section 4.3). The value of k will be given later. Parameter e_t controls trade-off between SfM and GPS in the fusion result. The smaller e_t, the stronger constraint enforced by SfM, the less tolerance for inaccurate GPS. If $e_t = e(\mathbf{x}^*)$, the fusion result $\mathbf{x}$ is equal to the pure SfM result $\mathbf{x}^*$: GPS is ignored. In this paper, we choose $e_t = 1.05^2 e(\mathbf{x}^*)$, *i.e.* a RMS increase of 5% is accepted for fusion. The other parameters of UBA and IBA are

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

These weights are such that the ratio between image term and GPS term in e_U (e_I, respectively) is 1 (10, respectively) before the fusion optimization.

Step (5) is used in the main loop once the SfM result is registered in the GPS coordinate system. The registration method is the following. First we select times $t_0 = 0$ and t_1 such that the distance between the two GPS positions is greater than 10 meters. Then we define the vertical direction in the SfM result assuming that both x-axis and motion of the camera are horizontal between t_0 and t_1. Now three points are defined in both coordinates systems (SfM and GPS) and a similarity transformation is estimated from these points. Finally, the SfM result is mapped in the GPS coordinate system using the similarity transformation.

5.2. Experimental conditions and notations

Our GPS and camera are mounted on a car. The car trajectory has straight lines and sharp curves, traffic circles, stop and go due to traffic lights. It is 4 km long. The scene includes low and high buildings, trees and moving vehicles.

The GPS is low cost (Ublox Antaris 4). It provides one 2D location (longitude, latitude) at 1Hz and the altitude is set to 0. Once the GPS coordinates are converted to euclidean coordinates in meters, linear interpolation is used to obtain a 3D GPS location at all times. The ground truth is provided by IXSEA LandINS (and RTK GPS, which is not low cost) at 10Hz. The mean, standard deviation and maximum of the low cost GPS error in our sequence are 4.28, 2.34, and 12.2, respectively (in meters).

The camera is monocular and calibrated; it points forward and provides 640×352 images (Fig. 1) at 25 Hz. 2480 keyframes are selected from 14850 images, such that there are about 600 Harris point matches between two consecutive keyframes. We assume that the distance between camera and GPS antenna is small in comparison to the GPS

Figure 1. Several images of the video sequence.

accuracy. The GPS coordinates of the camera ($\mathbf{x}_1^{gps}$) are therefore approximated by those of the GPS antenna.

The 3D location of keyframes are provided by six methods: SfM (GPS ignored), GPS (camera ignored), GT (ground truth), UBA, IBA and EBA. SfM, GPS and GT do not involve fusion. UBA, IBA and EBA have the same conditions for fusion: same keyframes, same matches, same maximum number of iterations ($It_{max} = 4$), and same k.

Let a and b two different methods that we would like to compare. Let l_a^i and e_a^i be the 3D location and the reprojection error (RMS) provided by method a at the i-th keyframe. We study the distribution of $\forall i, ||l_a^i - l_b^i||$, where $a \in \{\text{SfM}, \text{GPS}, \text{UBA}, \text{IBA}, \text{EBA}\}$, $b \in \{\text{GPS}, \text{GT}\}$. Its mean, standard deviation and maximum are m_a^b, σ_a^b and ∞_a^b. We also study the distribution of $\forall i, e_a^i/e_{\text{SfM}}^i$, where $a \in \{\text{UBA}, \text{IBA}, \text{EBA}\}$. Its mean, standard deviation and maximum are m_a^{2d}, σ_a^{2d} and ∞_a^{2d}. We refer to these distributions as "location errors" and "image errors", respectively. The distributions are estimated after the complete fusion of the sequence.

5.3. Comparison of UBA, IBA and EBA

Table 1 shows the location errors using $k = 40$. The three fusions (UBA, IBA, EBA) greatly reduce the errors relative to GPS to about 2 meters. The errors relative to ground truth are also greatly reduced to about 5 meters, which is the magnitude order of the GPS accuracy. However, the fusion methods are not able to improve the GPS accuracy since the fusion errors are slightly larger than the pure GPS errors. According to values of m_a^{GPS} and m_a^{GT}, the best results are obtained by IBA, followed by EBA and UBA. Figure 3 shows the evolution of location errors (relative to GPS) over time.

Figure 2 shows a local view of the 3D locations provided by the fusion BAs, in the case where there are high buildings at the road border. The car moves from right to left. On the left, we can see that the trajectory shapes of the fusion BAs are better than that of the GPS: fusion trajectories are smooth like GT trajectory, GPS trajectory (using linear

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

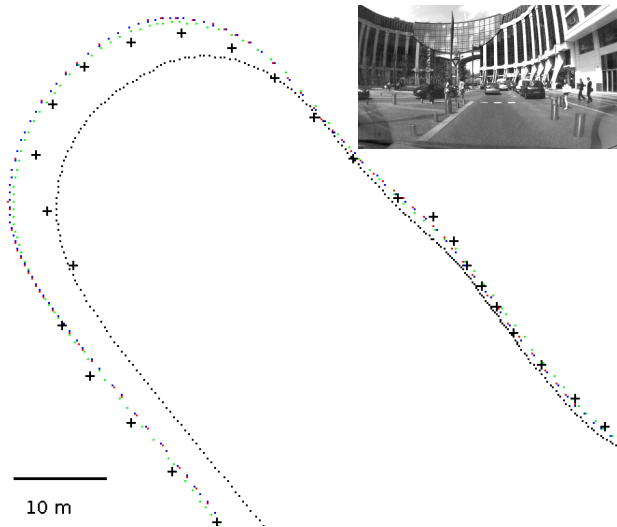Table 1. Location errors (in meters) for SfM, three local BAs for SfM-GPS fusion using $k = 40$, and pure GPS data.

Figure 2. Local view of the locations provided by UBA (red dots), IBA (green dots), EBA (blue dots), GPS (black crosses), GT (black dots). Red dots are superimposed by blue dots on the top-left corner (1 dot = 1 keyframe).

f	k=40			k=30		
	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. Image errors due to fusion.

interpolation) is not smooth at a point on the left. However, we can also see that the GPS does not provide a good (local) scale factor to the trajectory.

The image errors are shown on the left of Table 2 for $k = 40$. We check that their increases are acceptable for all fusion BAs since the ratios are close to 1.05, which is used to define threshold ϵ.

The mean times of U-/I-/EBA are 0.25, 0.27 and 0.28 seconds for each keyframe, respectively. These times are obtained with a core 2 duo 2.5Ghz laptop, sparse implementation of hessians, and Cholesky factorization of reduced camera system to solve the LM linear systems [14].

The same experiments are re-done using $k = 30$. We can

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. Location errors (in meters) of the camera for the fusion methods using $k = 30$.

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. Location and image errors of U-/IBA using $k = 40$.

see that the fusion is bad for UBA and IBA: the location errors are 10-40 larger than before (Table 3), and their curves are outside Figure 4 if the keyframe number is larger than a threshold. However, their image errors are acceptable (right of Table 2). The fusion is still correct for EBA: the location error (relative to GPS) is about 1.8 meter and the image error is slightly larger (about 1.07). Other experiments show that all fusion-BAs have location errors that are too large using $k = 25$.

5.4. Weight changes for UBA and IBA

Remember that UBA and IBA require choosing weights β and γ, respectively. So we re-do the UBA and IBA fusions using different weights around the default values in Eq. 22. The results are given in Table 4. We can see that the fusion results are similar if we divide or multiply the weights by 2. We can also see that large changes of weight (division or multiplication by 10) provide bad fusion results. These experiments suggest that the tuning of the weights is important, although it is not difficult to get weight which provides acceptable fusion results. Furthermore, they confirm that IBA provides the best 3D location results.

6. Conclusion

Two constrained bundle adjustments IBA and EBA were introduced to fuse GPS and Structure-from-Motion data. The former is constrained by inequality using penalty function, the latter involves equality constraint and is derived from a reduction method. Both BA algorithms are described in details. We also explain why the fusion difficulty increases when the number k of poses optimized in fusion BA decreases. Experiments compare our two BAs with the
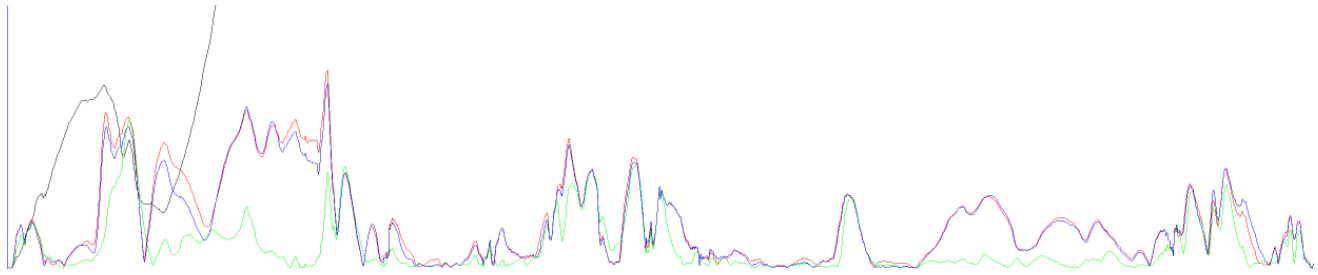
Figure 3. Location errors (relative to GPS) by SfM (black), UBA (red), IBA (green) and EBA (blue) using $k = 40$. The x-axis is the keyframe number in range 0-2479 and the y-axis is the location error in range 0-15 meters.
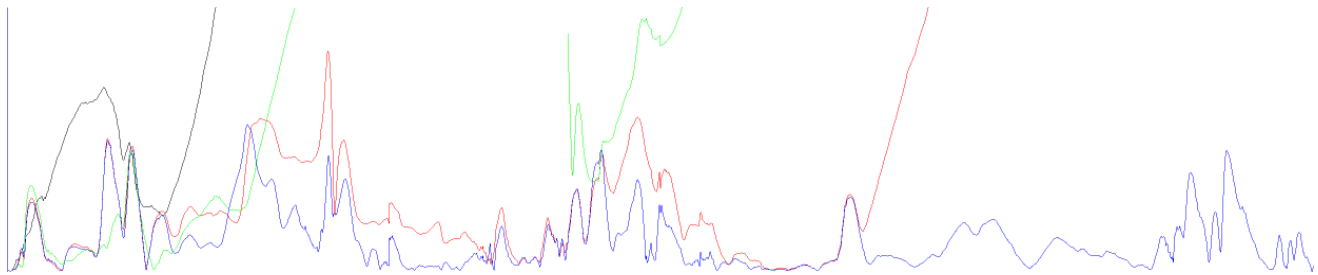

Figure 4. Location errors (relative to GPS) by SfM (black), UBA (red), IBA (green) and EBA (blue) using $k = 30$. The x-axis is the keyframe number in range 0-2479 and the y-axis is the location error in range 0-15 meters.

existing UBA (which minimizes a weighted sum of image and GPS errors), in the context of incremental Structure-from-Motion applied on a long urban image sequence.

The three fusion BAs greatly improve the poses of the Structure-from-Motion; the resulting increases of reprojection errors are small. According to ground truth, the resulting pose accuracies are similar to that of the GPS. The GPS accuracy is slightly better (it is the only sensor which provides absolute data, our monocular camera can not). IBA provides the best fusion results, EBA is ranked #2 but it has two advantages: it is the less sensitive method to a small k (the smaller the k, the faster the computation by local BA) and it does not require weight choice.

Future work includes experiments with other fusion BAs, improvement to initialize the visual reconstruction in the GPS coordinate system, comparison of the fusion BAs in the batch context, fusion with GPS providing the altitude or other sensors, application to georeferenced 3D modeling.

References

[1] T. Bailey and H. Durrant-Whyte. Simultaneous localisation and mapping (slam): Part i and ii. *IEEE Robotics and Automation Magazine*, 13(2), 2006. 3025

[2] Y. Bard. *Non-Linear Parameter Estimation*. Academic Press Inc. (London) LTD, 1974. 3026

[3] C. Ellum. Integration of raw gps measurements into a bundle adjustment. In *IAPRS series vol. XXXV*, 2006. 3025

[4] A. Eudes and M. Lhuillier. Error propagations for local bundle adjustment. In *CVPR'09*. 3025

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 3028

[6] Y. Jeong, D. Nister, D. Steedly, R. Szeliski, and I.S.Kweon. Pushing the envelope of modern methods for bundle adjustment. In *CVPR'10*. 3025

[7] H. Kume, T. Takemoti, T. Sato, and N. Yokoya. Extrinsic camera parameter estimation using video images and gps considering gps position accuracy. In *ICPR'10*. 3025

[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. In *CVPR'09*. 3025

[9] J. McGlone. *The Manual of Photogrammetry, Fifth Edition*. ISBN-10 1570830711, ASPRS, 2004. 3025, 3026

[10] J. Michot, A. Bartoli, and F. Gaspard. Bi-objective bundle adjustment with application to multi-sensor slam. In *3DPVT'10*. 3025

[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. 3025, 3029

[12] K. Ni, D. Steedly, and F. Dellaert. Out-of-core bundle adjustment for large-scale 3d reconstruction. In *ICCV'07*. 3025

[13] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipies in C, Second Edition*. Cambridge University Press, 1999. 3027

[14] B. Triggs, P. F. McLauchlan, R. Hartley, and A. W. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice*. 2000. 3025, 3026, 3027, 3028, 3029, 3031