HAL
open science

Incremental Fusion of Structure-from-Motion and GPS using Constrained Bundle Adjustment

Maxime Lhuillier

To cite this version:

Maxime Lhuillier. Incremental Fusion of Structure-from-Motion and GPS using Constrained Bundle Adjustment. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34 (12), pp.2489-2495. hal-01635480

HAL Id: hal-01635480

https://hal.science/hal-01635480

Submitted on 15 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Maxime Lhuillier

Institut Pascal, UMR 6602, CNRS/UBP/IFMA

24 avenue des Landais,

63177 Aubière Cedex, France.

Mail: Maxime.Lhuillier [AT] free.fr

Tel: +33(0)4 73 40 75 93

Fax: +33(0)4 73 40 72 62

http://maxime.lhuillier.free.fr

The reference of this paper is: Maxime Lhuillier, Incremental Fusion of Structure-from-Motion and GPS using Constrained Bundle Adjustments, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(12), 2012.

The published version of this paper is available at
http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=6332439

Copyright ©2012 IEEE

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

Maxime Lhuillier

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 long monocular image sequences and low cost GPS.

1 Introduction

Bundle adjustment (BA) is an iterative method of estimating camera poses and 3d points detected in an image sequence [12]. The resulting poses and points minimize a sum of squared reprojection errors.

Recent BA developments mainly concern accelerations for long sequences such as multicore BA [13], conjugated gradient [1], and local BA (LBA) [10].

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 terms weighted by measure covariances [8]. 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 [7]. Recent work combines GPS and image measures [5] (or inertial and image measures [9]) using LBA, which minimizes a weighted sum of GPS (or inertial) and image terms. In [9], several weights are experimented. In [5],

the experiments are limited to a small sequence (70 m) and the GPS term is defined by a high order polynomial.

In [6], new constrained BAs are introduced for SfM-GPS fusion. These BAs enforce an upper bound for the reprojection error, while the other fusion BAs [8, 9, 5] do not guarantee a small reprojection error and requires a weight. This paper also compares the results of the fusion BAs in a context which is useful for applications: the incremental SfM based on LBA [10]. In experiments, low cost GPS and monocular (calibrated) camera are mounted on a car moving in urban area. The trajectory length is larger (4 km) than in the previous works.

The current paper is an improved version of [6]. The prerequisites detail our assumptions (Section 2.1) and studies our upper bound-based fusion scheme in a simple case (Section 2.2): if the sum of squared reprojection errors is approximated by one quadratic Taylor expansion. Section 3 provides a brief overview of BAs which solve the SfM-GPS fusion problem. Only sparse Levenberg-Marquardt [12] (second order) based methods are considered here. Section 4 introduces our two constrained BAs for fusion, which involve inequality constraint. Section 5 provides the detailed algorithms (useful for re-implementers). Lastly, Section 6 shows experiments in the same context as [6].

The additional contributions over [6] includes Section 2.2 and new experiments. Section 2.2 provides interesting properties and helps to convince the reader that SfM-GPS fusion is possible without calculation of SfM covariance. The properties detail the link between the ϵ-indifference region [2] defined by our upper bound, the SfM covariance (that we do not estimate), and the GPS locations where fusion is possible.

The new experiments show the robustness of the fusion methods against several important factors: upper bounds for image error and track lengths, time shift between GPS and video recorders, frequency of GPS perturbations, number of iterations, incomplete GPS data. Lastly, a 5 km long sequence is experimented with GPS providing altitude (the GPS in [6] does not).

2 Prerequisites

Section 2.1 introduces notations and assumptions. Section 2.2 details our fusion scheme when the sum of squared reprojection errors is approximated by one quadratic Taylor expansion.

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). Vector $\mathbf{x}$ concatenates 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 fusion BA is the minimizer $\mathbf{x}^*$ of e, i.e. $\forall \mathbf{x}, e(\mathbf{x}^*) \leq e(\mathbf{x})$. Standard BA (not fusion BA) provides $\mathbf{x}^*$.

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 $\mathtt{P} = \begin{pmatrix}\mathtt{I} & 0\end{pmatrix}$ be such that $\mathbf{x}_1 = \mathtt{P}\mathbf{x}$.

Let $\mathbf{x}_1^{gps}$ be the location(s) of the camera provided by GPS at the same time(s) as $\mathbf{x}_1$. Assuming that the GPS drift (or accumulation error) is bounded and that of SfM is not, the ideal output $\mathbf{x}$ of fusion BA meet $\mathbf{x}_1 \approx \mathbf{x}_1^{gps}$.

Vector $\mathbf{x}_2$ concatenates all 3d points, all rotations of the camera, and the camera locations without GPS data.

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

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

Last, we assume that $\mathtt{H} > 0$, i.e. the hessian of e is positive definite in a neighborhood of $\mathbf{x}^*$.

2.2 Quadratic Taylor approximation

Let q and $\mathtt{H}^*$ be the quadratic Taylor expansion and Hessian of e at $\mathbf{x}^*$. Since e gradient is zero at $\mathbf{x}^*$,

$$e(\mathbf{x}^* + \mathbf{\Delta}) \approx q(\mathbf{\Delta}) = e(\mathbf{x}^*) + 0.5\mathbf{\Delta}^T\mathtt{H}^*\mathbf{\Delta}. \tag{2}$$

In the paper we use block-wise notations

$$\begin{pmatrix}\mathbf{x}_1^* & \mathbf{x}_1 & \mathbf{\Delta}_1 \\ \mathbf{x}_2^* & \mathbf{x}_2 & \mathbf{\Delta}_2\end{pmatrix} = \begin{pmatrix}\mathbf{x}^* & \mathbf{x} & \mathbf{\Delta}\end{pmatrix}, \begin{pmatrix}\mathtt{H}_1 & \mathtt{H}_{21}^T \\ \mathtt{H}_{21} & \mathtt{H}_2\end{pmatrix} = \mathtt{H}^*. \tag{3}$$

Assume that $\mathbf{\Delta}_1$ is a step of $\mathbf{x}_1^*$ to remove (or to reduce) the SfM drift, i.e. $\mathbf{x}_1^* + \mathbf{\Delta}_1 = \mathbf{x}_1^{gps}$. Now we should find $\mathbf{x}$ such that both Eq. 1 and $\mathbf{x}_1 = \mathbf{x}_1^* + \mathbf{\Delta}_1$ are meet.

The set of values $\mathbf{x}$ defined by Eq. 1 is called ϵ-indifference region ([2], p.171) where $\epsilon = e_t - e(\mathbf{x}^*)$. Since ϵ is small and $\mathtt{H}^* > 0$, Eq. 2 is used to approximate the ϵ-indifference region by ellipsoïd ([2], p.172)

$$E_{\mathbf{x}^*}^{\epsilon} = \{\mathbf{x}^* + \mathbf{\Delta}, q(\mathbf{\Delta}) \leq e(\mathbf{x}^*) + \epsilon\}. \tag{4}$$

Now we should find $\mathbf{x} \in E_{\mathbf{x}^*}^{\epsilon}$ such that $\mathbf{x}_1 = \mathbf{x}_1^* + \mathbf{\Delta}_1$. In other words, $\mathbf{x}_1^{gps}$ should be in the ellipsoïd projection

$$\begin{aligned} E_{\mathbf{x}_1^*}^{\epsilon} &= \{\mathbf{x}_1, \exists \mathbf{x}_2, \begin{pmatrix}\mathbf{x}_1^T & \mathbf{x}_2^T\end{pmatrix}^T \in E_{\mathbf{x}^*}^{\epsilon}\} \\ &= \{\mathbf{x}_1^* + \mathbf{\Delta}_1, \exists \mathbf{\Delta}_2, q(\begin{pmatrix}\mathbf{\Delta}_1 \\ \mathbf{\Delta}_2\end{pmatrix}) \leq e(\mathbf{x}^*) + \epsilon\}. \end{aligned} \tag{5}$$

Lemma 1 is useful to explicit $E_{\mathbf{x}_1^*}^{\epsilon}$.

Lemma 1: *Function* $\mathbf{\Delta}_2 \mapsto q(\begin{pmatrix}\mathbf{\Delta}_1^T & \mathbf{\Delta}_2^T\end{pmatrix}^T)$ *has minimum*

$$q(\begin{pmatrix}\mathbf{\Delta}_1 \\ -\mathtt{H}_2^{-1}\mathtt{H}_{21}\mathbf{\Delta}_1\end{pmatrix}) = e(\mathbf{x}^*) + 0.5\mathbf{\Delta}_1^T\mathtt{C}_1^{-1}\mathbf{\Delta}_1, \tag{6}$$

where $\mathtt{C}_1 = (\mathtt{H}_1 - \mathtt{H}_{21}^T \mathtt{H}_2^{-1} \mathtt{H}_{21})^{-1}$ *is the top-left block of* $\mathtt{H}^{*-1}$.

Proof: Thanks to $\mathtt{H}^* > 0$ and Section 6.1 of [12], $\mathtt{H}_1 - \mathtt{H}_{21}^T \mathtt{H}_2^{-1} \mathtt{H}_{21}$ is the Schur complement of $\mathtt{H}_2$ in $\mathtt{H}^*$ and $\mathtt{C}_1$ is the top-left block of $\mathtt{H}^{*-1}$. Furthermore, $\mathtt{H}^* > 0$ implies $\mathtt{H}_2 > 0$. Thus, quadratic function $\mathbf{\Delta}_2 \mapsto q(\begin{pmatrix} \mathbf{\Delta}_1^T & \mathbf{\Delta}_2^T \end{pmatrix}^T)$ has minimizer $-\mathtt{H}_2^{-1} \mathtt{H}_{21} \mathbf{\Delta}_1$ and minimum

$$e(\mathbf{x}^*) + 0.5 \mathbf{\Delta}_1^T \begin{pmatrix} \mathtt{I} \\ -\mathtt{H}_2^{-1} \mathtt{H}_{21} \end{pmatrix}^T \mathtt{H}^* \begin{pmatrix} \mathtt{I} \\ -\mathtt{H}_2^{-1} \mathtt{H}_{21} \end{pmatrix} \mathbf{\Delta}_1. \tag{7}$$

□

Thanks to Lemma 1, we can use $\mathbf{\Delta}_2 = -\mathtt{H}_2^{-1} \mathtt{H}_{21} \mathbf{\Delta}_1$ in Eq. 5 and obtain

$$E_{\mathbf{x}_1^*}^{\epsilon} = \{\mathbf{x}_1^* + \mathbf{\Delta}_1, \mathbf{\Delta}_1^T \mathtt{C}_1^{-1} \mathbf{\Delta}_1 \leq 2\epsilon\}. \tag{8}$$

Theorem 1 summarizes the derivations of Section 2.2.

Theorem 1: *Thanks to the quadratic Taylor approximation of* e *at* $\mathbf{x}^*$*, the fusion problem defined by*

$$e(\mathbf{x}) \leq e_t \text{ and } \mathbf{x}_1 = \mathbf{x}_1^{gps} \tag{9}$$

has solution(s) $\mathbf{x}$ *if and only if* $\mathbf{x}_1^{gps}$ *is in the ellipsoïd* $E_{\mathbf{x}_1^*}^{\epsilon}$ *(Eq. 8) where* $\epsilon = e_t - e(\mathbf{x}^*)$ *and* $\mathtt{C}_1 = (\mathtt{H}_1 - \mathtt{H}_{21}^T \mathtt{H}_2^{-1} \mathtt{H}_{21})^{-1}$*. In this case,* $e(\mathbf{x})$ *is minimized by choosing*

$$\mathbf{x}_2 = \mathbf{x}_2^* - \mathtt{H}_2^{-1} \mathtt{H}_{21} (\mathbf{x}_1^{gps} - \mathbf{x}_1^*). \tag{10}$$

If $\mathbf{x}_1^{gps} \notin E_{\mathbf{x}_1^*}^{\epsilon}$, Theorem 1 can still be used to fuse SfM and GPS incompletely: we replace $\mathbf{x}_1^{gps}$ by $\tilde{\mathbf{x}}_1^{gps} \in E_{\mathbf{x}_1^*}^{\epsilon}$ such that $\tilde{\mathbf{x}}_1^{gps}$ is close as possible to $\mathbf{x}_1^{gps}$.

Lastly, we provide a probabilistic interpretation of $\mathtt{C}_1$ and $E_{\mathbf{x}_1^*}^{\epsilon}$. Remind that $\mathbf{x}^*$ is the minimizer of sum of squared reprojection errors. Under the assumption that the image noise follows the zero-mean normalized Gaussian vector, the $\mathbf{x}^*$ covariance is approximated by $\mathtt{H}^{*-1}$ [4]. Then we see that $\mathtt{C}_1$ is the covariance matrix of $\mathbf{x}_1^*$ and $E_{\mathbf{x}_1^*}^{\epsilon}$ is an uncertainty ellipsoïd of $\mathbf{x}_1^*$.

3 BA candidates for SfM-GPS fusion

Here we review three BAs which fuse SfM and GPS. They meet the requirements of Section 2.1.

3.1 UBA: BA without explicit constraint

Such a BA was used to combine measurements from different sensors [8]. We refer to it as **UBA** or "unconstrained BA". A sum of weighted terms is minimized:

$$e_U(\mathbf{x}) = e(\mathbf{x}) + \beta||\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}||^2. \tag{11}$$

Here the problems are the adequate choice of weight β and the risk of inlier loss due to the term $\beta||\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}||^2$. 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. In our framework, the UBA output is ignored if $e(\mathbf{x}) > e_t$

3.2 IBA: BA with inequality constraint

Another method uses penalty function ([2], p.141). In our context, the iterations of this constrained BA enforce the inequality constraint in Eq. 1, i.e. $c_I(\mathbf{x}) > 0$ where $c_I(\mathbf{x}) = e_t - e(\mathbf{x})$. Here we minimize

$$e_I(\mathbf{x}) = \gamma/c_I(\mathbf{x}) + ||\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}||^2 \tag{12}$$

where $\gamma > 0$. Function $\mathbf{x} \mapsto ||\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}||$ 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 $\mathbf{x} \mapsto ||\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}||^2$ elsewhere.

Although the principle is simple, such an **IBA** was not used before for fusion of SfM and another sensor.

3.3 EBA: BA derived from equality constraints

BAs in [12] minimize $e(\mathbf{x})$ subject to equality constraint $c(\mathbf{x}) = 0$. At first glance, we could try $c(\mathbf{x}) = \mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}$ since we would like $\mathbf{x}_1 \approx \mathbf{x}_1^{gps}$.

One iteration improves $\mathbf{x}$ by adding step $\mathbf{\Delta}$ subject to the linearized constraint $c(\mathbf{x} + \mathbf{\Delta}) \approx c(\mathbf{x}) + \frac{\partial c}{\partial \mathbf{x}}\mathbf{\Delta} = \mathbf{0}$. 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 $||c(\mathbf{x})|| = ||\frac{\partial c}{\partial \mathbf{x}}\mathbf{\Delta}||$.

Now we see that $c(\mathbf{x}) = \mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}$ can not be used: on the one hand the constrained BAs in [12] require small $||c(\mathbf{x})||$, while on the other $c(\mathbf{x}^*) = \mathbf{x}_1^* - \mathbf{x}_1^{gps}$ may have large modulus since it is the drift between SfM and GPS.

Therefore we introduce **EBA**, which is derived from a constrained BA in [12]. EBA is a different method and c is replaced by another function

$$c_\alpha(\mathbf{x}) = \mathtt{P}\mathbf{x} - ((1-\alpha)\mathbf{x}_1^{gps} + \alpha\mathbf{x}_1^*) \text{ where } \alpha \in [0,1]. \tag{13}$$

Note that Eq. 13 is the same as Eq. 14 in [6], i.e. $c_\alpha(\mathbf{x}) = c(\mathbf{x}) - \alpha c(\mathbf{x}^*)$. Eq. 13 makes easier the understanding of $c_\alpha(\mathbf{x}) = 0$: $\mathbf{x}_1$ is a linear interpolation of $\mathbf{x}_1^{gps}$ and $\mathbf{x}_1^*$.

Eq. 13 implies $c_1(\mathbf{x}^*) = \mathbf{0}$ and $c_0(\mathbf{x}) = c(\mathbf{x})$. EBA decreases α progressively 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). A decrease of α may produce an increase of $e(\mathbf{x})$, but this increase is moderated since we integrate in EBA the reduction method (a constrained BA in Section 4.4 of [12]). This is useful to meet Eq. 1.

Note that EBA minimizes α and the (integrated) reduction method minimizes $e(\mathbf{x})$. For the paper clarity, the reduction method and EBA are described in two different Sections 4.3 and 4.4.

4 Iteration of BAs

Section 4 describes the iterations of Levenberg-Marquardt (LM), IBA and EBA (the former is useful to explain the latters). The supplementary material shows that successful iteration is possible in all cases.

The quadratic Taylor expansion of e at $\mathbf{x}$ is

$$e(\mathbf{x} + \mathbf{\Delta}) \approx e(\mathbf{x}) + \mathbf{g}^T \mathbf{\Delta} + 0.5 \mathbf{\Delta}^T \mathtt{H} \mathbf{\Delta} \tag{14}$$

where $\mathbf{g}$ and $\mathtt{H}$ are the gradient and hessian of e. 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 $\mathbf{g} = 2\mathtt{J}^T E(\mathbf{x})$ and use the Gauss-Newton approximation $\mathtt{H} \approx 2\mathtt{J}^T\mathtt{J}$. We assume $\mathtt{J}^T\mathtt{J} > 0$ since $\mathtt{H} > 0$ (Section 2.1).

4.1 Levenberg-Marquardt without constraint

The LM iteration to minimize $e(\mathbf{x})$ without constraint is the following [11] (UBA minimizes a different function using LM). Efficient sparse methods are used to solve $(\mathtt{H} + \lambda \mathrm{diag}(\mathtt{H}))\mathbf{\Delta} = -\mathbf{g}$ for the current value of $\mathbf{x}$ and a damping coefficient $\lambda > 0$. If $e(\mathbf{x} + \mathbf{\Delta}) < e(\mathbf{x})$, the iteration is successful: $\mathbf{x}$ is replaced by $\mathbf{x} + \mathbf{\Delta}$ and λ is replaced by $\lambda/10$. Otherwise, λ is replaced by 10λ.

4.2 IBA

The method is the same as in Section 4.1, except for the calculation of $\mathbf{\Delta}$. Let x_i be a coefficient of $\mathbf{x}$ and $f(\mathbf{x}) = \gamma/(e_t - e(\mathbf{x}))$. We have

$$\begin{aligned} \frac{\partial f}{\partial x_i} &= \frac{\gamma}{(e_t - e)^2} \frac{\partial e}{\partial x_i} \\ \frac{\partial^2 f}{\partial x_i \partial x_j} &= \frac{\gamma}{(e_t - e)^3} ((e_t - e) \frac{\partial^2 e}{\partial x_i \partial x_j} + 2 \frac{\partial e}{\partial x_i} \frac{\partial e}{\partial x_j}). \end{aligned} \tag{15}$$

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:

$$\begin{aligned} \mathbf{g}_I &= \frac{\gamma}{(e_t - e)^2}\mathbf{g} + 2\mathtt{P}^T(\mathtt{P}\mathbf{x} - \mathbf{x}_1^{gps}) \\ \mathtt{H}_I &\approx \frac{2\gamma}{(e_t - e)^3}((e_t - e)\mathtt{J}^T\mathtt{J} + \mathbf{g}\mathbf{g}^T) + 2\mathtt{P}^T\mathtt{P}. \end{aligned} \tag{16}$$

Now, the linear system $(\mathtt{H}_I + \lambda\mathrm{diag}(\mathtt{H}_I))\mathbf{\Delta} = -\mathbf{g}_I$ is solved. This can not be solved as in Section 4.1 since $\mathtt{H}_I$ is not sparse due to the dense term $\mathbf{g}\mathbf{g}^T$. Section 5.1 provides an efficient method to solve this linear system.

4.3 Reduction Method (BA with equality constraint)

Now the LM iteration to minimize $e(\mathbf{x})$ subject to constraint $c(\mathbf{x}) = 0$ is described [12]. We use notations

$$\begin{pmatrix} \mathbf{x}_1 & \mathbf{\Delta}_1 & \mathbf{g}_1 \\ \mathbf{x}_2 & \mathbf{\Delta}_2 & \mathbf{g}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{x} & \mathbf{\Delta} & \mathbf{g} \end{pmatrix}, \begin{pmatrix} \mathtt{H}_1 & \mathtt{H}_{21}^T \\ \mathtt{H}_{21} & \mathtt{H}_2 \end{pmatrix} = \mathtt{H} \tag{17}$$

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 $\mathbf{\Delta}$ is such that

$$c(\mathbf{x} + \mathbf{\Delta}) \approx c(\mathbf{x}) + \mathtt{C}_1\mathbf{\Delta}_1 + \mathtt{C}_2\mathbf{\Delta}_2 = c(\mathbf{x}) + \mathbf{\Delta}_1 = \mathbf{0}. \tag{18}$$

Then, $\mathbf{\Delta}$ is

$$\Delta(\mathbf{\Delta}_2) = \begin{pmatrix} -c(\mathbf{x})^T & \mathbf{\Delta}_2^T \end{pmatrix}^T. \tag{19}$$

Thanks to Eq. 14 and $\mathbf{\Delta} = \Delta(\mathbf{\Delta}_2)$, we obtain

$$e(\mathbf{x} + \Delta(\mathbf{\Delta}_2)) \approx \bar{e}_2 + \mathbf{\Delta}_2^T\bar{\mathbf{g}}_2 + 0.5\mathbf{\Delta}_2^T\mathtt{H}_2\mathbf{\Delta}_2 \tag{20}$$

where

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

Step $\mathbf{\Delta}_2$ meets $(\mathtt{H}_2 + \lambda\mathrm{diag}(\mathtt{H}_2))\mathbf{\Delta}_2 = -\bar{\mathbf{g}}_2$. Now the iteration is the same as in Section 4.1 using $\mathbf{\Delta} = \Delta(\mathbf{\Delta}_2)$.

4.4 From Reduction Method to EBA

Assume that EBA is the reduction method using the constraint in Eq. 13. A problem is the descending condition $e(\mathbf{x} + \Delta(\mathbf{\Delta}_2)) < e(\mathbf{x})$ to test step $\mathbf{\Delta} = \Delta(\mathbf{\Delta}_2)$. 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 very beginning of EBA. However, we remind that our condition for fusion is Eq. 1. We solve this problem, substituting the descending condition by

$$e(\mathbf{x} + \Delta(\mathbf{\Delta}_2)) < e_t. \tag{22}$$

Now the $\mathbf{\Delta}_2$ calculation in a successful EBA iteration is concisely written as: find positive δ and λ such that

$$\begin{cases} \bar{\mathbf{g}}_2 = \mathbf{g}_2 - \mathtt{H}_{21} c_{\alpha-\delta}(\mathbf{x}) \\ (\mathtt{H}_2 + \lambda \mathrm{diag}(\mathtt{H}_2))\mathbf{\Delta}_2 = -\bar{\mathbf{g}}_2 \\ e(\mathbf{x} + \begin{pmatrix} -c_{\alpha-\delta}(\mathbf{x})^T & \mathbf{\Delta}_2^T \end{pmatrix}^T) < e_t. \end{cases} \tag{23}$$

Then we add $\mathbf{\Delta} = \begin{pmatrix} -c_{\alpha-\delta}(\mathbf{x})^T & \mathbf{\Delta}_2^T \end{pmatrix}^T$ to $\mathbf{x}$, and subtract δ from α. The detailed algorithm is in Section 5.2. According to the supplementary material, Eqs. 23 have a solution thanks to small enough δ and large enough λ.

5 Implementation

Now we will explain how to implement efficiently IBA (Section 5.1) and EBA (Section 5.2).

5.1 IBA

In Section 4.2, $(\mathtt{H}_I + \lambda \mathrm{diag}(\mathtt{H}_I))\mathbf{\Delta} = -\mathbf{g}_I$ should be solved efficiently. Let $\tilde{\mathtt{H}}$ and $\tilde{\mathbf{g}}$ be such that

$$\mathtt{H}_I + \lambda \mathrm{diag}(\mathtt{H}_I) = \tilde{\mathtt{H}} + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T, \tilde{\mathbf{g}} = \sqrt{\frac{2\gamma}{(e_t - e)^3}}\mathbf{g}. \tag{24}$$

Basic computation shows that

$$(\tilde{\mathtt{H}} + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T)^{-1} = (\mathtt{I} - \frac{\tilde{\mathtt{H}}^{-1}\tilde{\mathbf{g}}\tilde{\mathbf{g}}^T}{1 + \tilde{\mathbf{g}}^T\tilde{\mathtt{H}}^{-1}\tilde{\mathbf{g}}})\tilde{\mathtt{H}}^{-1}. \tag{25}$$

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

$$\mathbf{\Delta} = -(\tilde{\mathtt{H}} + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T)^{-1}\mathbf{g}_I = \mathbf{a} - \frac{\tilde{\mathbf{g}}^T\mathbf{a}}{1 + \tilde{\mathbf{g}}^T\mathbf{b}}\mathbf{b}. \tag{26}$$

Now we explain how to estimate $\mathbf{a}$ and $\mathbf{b}$. According to Eqs. 16 and 24, $\tilde{\mathtt{H}}$ has the sparse structure of $\mathtt{J}^T\mathtt{J}$. More precisely [4], we have $\tilde{\mathtt{H}} = \begin{pmatrix} \mathtt{U} & \mathtt{W} \\ \mathtt{W}^T & \mathtt{V} \end{pmatrix}$ and $\tilde{\mathtt{H}} > 0$ where $\mathtt{U}$ is a 6×6 block-wise matrix, $\mathtt{V}$ is a 3×3 block-wise invertible diagonal matrix, and $\mathtt{W}$ 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 $\tilde{\mathtt{H}}\mathbf{a} = -\mathbf{g}_I$ and $\tilde{\mathtt{H}}\mathbf{b} = \tilde{\mathbf{g}}$ are solved using the same efficient sparse method [4] as the linear system $(\mathtt{H} + \lambda \mathrm{diag}(\mathtt{H}))\mathbf{\Delta} = -\mathbf{g}$.

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. The output is $\mathbf{x}$ such that $e(\mathbf{x}) < e_t$ and $e_I(\mathbf{x})$ has the smallest possible value.

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

5.2 EBA

Solving the linear system of Eqs. 23 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 $\mathbf{\Delta}_2^a$ and $\mathbf{\Delta}_2^b$ such that

$$(\mathtt{H}_2 + \lambda\mathrm{diag}(\mathtt{H}_2))\begin{pmatrix}\mathbf{\Delta}_2^a & \mathbf{\Delta}_2^b\end{pmatrix} = \begin{pmatrix}-\mathbf{g}_2 & \mathtt{H}_{21}\end{pmatrix} \tag{27}$$

and obtain $\mathbf{\Delta}_2 = \mathbf{\Delta}_2^a + \mathbf{\Delta}_2^b c_{\alpha-\delta}(\mathbf{x})$. Now we see the improvement: once the linear system in Eq. 27 is solved, $\mathbf{\Delta}_2$ is obtained very efficiently for all tried δ.

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 $\mathbf{\Delta}^T = \begin{pmatrix}\mathbf{0}^T & (\mathbf{\Delta}_2^a)^T\end{pmatrix}$. Then we find λ such that $e(\mathbf{x} + \mathbf{\Delta}) < e(\mathbf{x})$ as in unconstrained BA (U-iteration).

Remind that EBA minimizes α, but it is interesting to obtain the smallest $e(\mathbf{x})$ for a given α. Thus, we alternate successful iteration with $\delta > 0$ (E-iteration) and successful U-iteration to decrease e as much as possible. The U-iterations do not update α. 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. The output is $(\mathbf{x}, \alpha)$ such that $e(\mathbf{x}) < e_t$, $\mathbf{x}_1 = (1-\alpha)\mathbf{x}_1^{gps} + \alpha\mathbf{x}_1^*$ and the smallest α as possible.

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

6 Experiments

6.1 Integrating fusion to LBA-based SfM

SfM [10] 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 $N - n = 7$ [10].

Our paper adds step (5), a fusion step which is the local version of UBA, IBA or EBA: e 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 LBA using a single iteration of standard LBA. Vector $\mathbf{x}_1$ is the 3d location of the most recent key-frame. Fusion LBA does not involve point outliers since they are rejected by steps (3-4) as in [10].

UBA, IBA and EBA run under the same conditions: same keyframes, same matches, same maximum number of iterations ($It_{max} = 4$), same k and e_t. Our default values are $k = 40$ and $e_t = 1.05^2 e(\mathbf{x}^*)$, i.e. a RMS increase of 5% is accepted for fusion. The other (default) parameters of UBA and IBA are

$$\beta = \frac{e(\mathbf{x}^*)}{||\mathtt{P}\mathbf{x}^* - \mathbf{x}_1^{gps}||^2},\ \gamma = \frac{e_t - e(\mathbf{x}^*)}{10}||\mathtt{P}\mathbf{x}^* - \mathbf{x}_1^{gps}||^2. \tag{28}$$

These weights are such that the ratio between image term and GPS term in e_U (e_I, respectively) is 1 (0.1, 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 coordinate 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.

6.2 Notations

The 3d location of keyframes are provided by six methods: SfM, GPS, GT (ground truth), UBA, IBA and EBA.

Let a and b be two different methods that we would like to compare. Let $\mathbf{l}_a^i$ 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 $\forall i, ||\mathbf{l}_a^i - \mathbf{l}_b^i||$, where $a \in \{\mathrm{SfM}, \mathrm{GPS}, \mathrm{UBA}, \mathrm{IBA}, \mathrm{EBA}\}$, $b \in \{\mathrm{GPS}, \mathrm{GT}\}$. Its mean, standard deviation and maximum are m_a^b, σ_a^b and ∞_a^b in meters. We also study the distribution of $\forall i, e_a^i / e_{\mathrm{SfM}}^i$, where $a \in \{\mathrm{UBA}, \mathrm{IBA}, \mathrm{EBA}\}$. 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. Here $\mathbf{l}_a^i$ and e_a^i are estimated after the calculation of the entire sequence by method a.

6.3 Experimental conditions for sequence 1

Our GPS and camera are mounted on a car. Its trajectory has straight lines, sharp curves, traffic circles, stop and go due to traffic lights. It is 4 km long.

Figure 1: Images of sequence 1.

f	m_f^{gps}	σ_f^{gps}	∞_f^{gps}	m_f^{gt}	σ_f^{gt}	∞_f^{gt}	m_f^{2d}	σ_f^{2d}
SfM	165	172	591	164	172	592	1	0
UBA	2.61	2.40	11.3	5.59	3.18	14.0	1.04	.044
IBA	1.24	1.50	8.47	4.57	2.83	12.1	1.05	.046
EBA	2.48	2.27	10.5	5.49	3.12	14.0	1.04	.045
GPS	0	0	0	4.28	2.34	12.2	-	-

Table 1: Location errors and image errors using the default parameters. If $f \in \{\text{UBA}, \text{IBA}, \text{EBA}\}$, $\infty_f^{2d} \in [1.28, 1.3]$.

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 at 10Hz by IXSEA LandINS and RTK (not low cost) GPS. We have $m_{gps}^{gt} = 4.28$, $\sigma_{gps}^{gt} = 2.34$ and $\infty_{gps}^{gt} = 12.2$, so the name "low cost GPS" is confirmed.

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 400 Harris point matches between three consecutive keyframes. We assume that the distance between camera and GPS antenna is small in comparison to GPS accuracy: the GPS coordinates of the camera ($\mathbf{x}_1^{gps}$) are approximated by those of the GPS antenna.

6.4 Comparison of UBA, IBA and EBA

Here we compare the methods using the default parameters in Section 6.1. Tab. 1 shows the location errors. 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 mean of 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. Tab. 1 also shows the image errors. We check that they are acceptable for all fusion BAs since they show

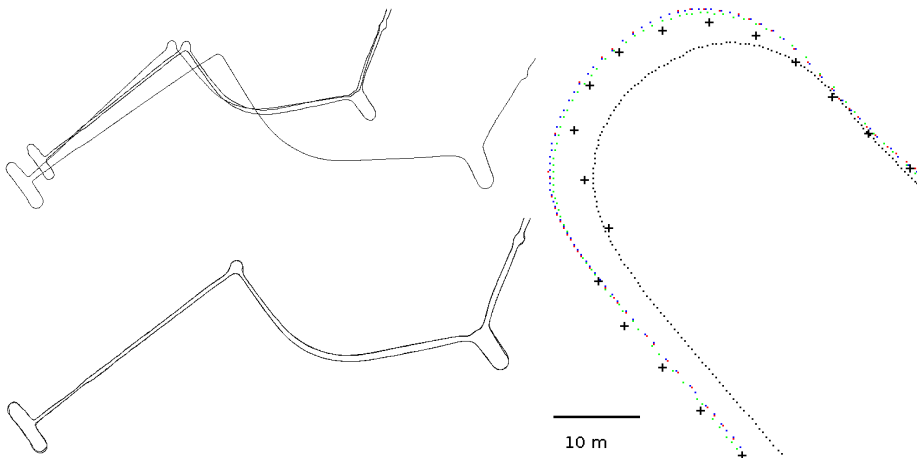

Figure 2: Top views of trajectories: GPS+SfM (top left), GPS+IBA (bottom left). Local view (right) of GPS (black crosses), GT (black dots), UBA (red dots), IBA (green dots), EBA (blue dots). One dot is one keyframe.

that the increase of RMS reprojection errors per keyframe due to fusion is about 5%. RMS e^i_{SfM} ranges from 0.37 to 0.54 pixel (the mean is 0.44), which implies that $e^i_f \leq 0.702$ pixel $\forall f, \forall i$.

Fig. 2 shows top views of GPS, SfM and IBA trajectories. We see the drift of SfM compared to GPS (top left). At this scale, it is difficult to see a difference between GPS and IBA (bottom left). The same observation can be done for UBA and EBA. Fig. 2 also shows a local view of the 3d locations provided by the fusion BAs (right), in the case where there are high buildings at the road border. The car moves from right to left. We 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 interpolation) is not smooth at a point on the left. We can also see that the GPS does not provide a good local scale factor to the trajectory.

The mean times of U-/I-/EBA are 0.25, 0.27 and 0.28 seconds for each keyframe, respectively. Here we use a core 2 duo 2.5Ghz laptop, sparse implementation of hessians, and Cholesky factorization of reduced camera system to solve the LM linear systems [12].

6.5 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 of Section 6.4 using different weights around the default values in Eq. 28. The results are given in Tab. 2. 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)

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 2: Location and image errors for weight changes.

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 has the best 3d location results.

6.6 Changes of sliding window size

We redo the experiments of Section 6.4 for different values of k. Tab. 3 shows that fusion is more difficult for small k. UBA, IBA and EBA provides acceptable results if $k \geq 40$: the mean of location errors is less than 6.07 and that of image errors is less than 1.05. Fusion is more difficult if $k < 40$, especially for UBA and IBA whose locations errors increases dramatically. For EBA, the location errors are acceptable (except $\infty_{eba}^{gt} = 35.1$ if $k = 25$), but the mean of image errors increase up to 1.098. Since small k is better for computation time, we found $k = 40$ is a good compromise for the 3 fusion BAs.

Such results are suggested by Section 2.2. Remind that the gauge is fixed at the beginning and $\mathbf{x}_1$ is at the end of the k most recent keyframes. In this context, we can assume that covariance $\mathtt{C}_1$ of $\mathbf{x}_1^*$ (Section 2.2) increases if the optimized sequence length k increases. Then Theorem 1 implies that $E_{\mathbf{x}_1^*}^{\epsilon}$, the region of the GPS locations where fusion is possible, increases if k increases. Thus, small k makes fusion more difficult.

6.7 Changes of image upper bound

We redo the experiments of Section 6.4 for different values of e_t. Here the notation $\mu = \sqrt{e_t/e(\mathbf{x}^*)}$ is more convenient. According to Section 6.1, the default value is $\mu = 1.05$. Tab. 3 shows that fusion is more difficult for small μ (or e_t): the location errors increase if μ decreases. This confirms the following intuition: the smaller e_t, the stronger constraint enforced by SfM, the less tolerance for inaccurate GPS. Furthermore, the means of image errors are less than 1.053. This might be surprising since they are cases where the image errors are greater than μ. We should remember that e_t is an upper bound for a sum of reprojec-

	m^{gt}_{uba}	m^{gt}_{iba}	m^{gt}_{eba}	m^{2d}_{uba}	m^{2d}_{iba}	m^{2d}_{eba}	∞^{2d}_{*}
default	5.59	4.57	5.49	1.037	1.049	1.038	1.301
k=25	54.0	195	6.62	1.013	1.009	1.098	1.439
k=30	65.2	43.3	4.99	1.018	1.021	1.070	1.501
k=35	34.6	33.5	5.15	1.035	1.032	1.051	1.392
k=45	5.84	4.56	5.74	1.029	1.042	1.032	1.291
k=50	6.07	4.59	5.86	1.027	1.035	1.028	1.263
μ=1.01	112	76.3	6.16	1.003	1.004	1.052	1.288
μ=1.03	94.9	5.01	5.40	1.022	1.049	1.039	1.314
μ=1.07	5.58	4.55	5.51	1.037	1.053	1.038	1.299
δt=-2	89.9	229	396	1.025	1.021	1.054	1.394
δt=-1	8.68	5.49	5.84	1.063	1.081	1.074	1.501
δt=-.5	5.39	4.87	5.59	1.058	1.061	1.052	1.456
δt=.5	5.91	4.71	5.79	1.032	1.048	1.035	1.283
δt=1	6.40	5.39	6.50	1.030	1.054	1.039	1.337
δt=2	10.3	7.64	7.99	1.060	1.071	1.073	1.444
δt=3	20.9	16.1	9.56	1.048	1.056	1.100	1.569
L_∞=6	5.39	4.57	5.28	1.041	1.059	1.043	1.320
L_∞=7	5.37	4.67	5.10	1.045	1.067	1.047	1.343
L_∞=8	11.9	9.47	4.99	1.035	1.045	1.048	1.360
L_∞=9	54.4	48.5	5.09	1.015	1.020	1.055	1.368
L_∞=∞	54.5	54.4	4.45	1.005	1.007	1.036	1.277
Itmx=1	85.1	120	4.61	1.015	1.027	1.057	1.312
Itmx=2	4.81	4.61	4.61	1.041	1.059	1.050	1.366
Itmx=3	5.49	4.56	4.83	1.038	1.051	1.042	1.306

Table 3: Location and image errors for parameter changes.

p	m_{uba}^{sfm}	m_{iba}^{sfm}	m_{eba}^{sfm}	m_{uba}^{gps}	m_{iba}^{gps}	m_{eba}^{gps}	∞_*^{2d}
10	4.56	4.04	1.31	10.7	11.0	9.95	1.48
100	4.56	4.04	1.31	11.9	14.5	10.1	1.64
200	442	11.4	10.2	443	10.4	9.31	1.48
400	10.7	12.3	10.1	6.43	7.93	4.67	1.48
800	10.3	12.4	9.94	3.91	6.62	2.17	1.48
1600	9.91	10.4	9.88	1.57	1.46	0.76	1.48

Table 4: Location and image errors if $i \mapsto \mathbf{l}_{gps}^i - \mathbf{l}_{sfm}^i$ has period p.

tion errors over the k most recent keyframes; it does not enforce upper bound for individual keyframe. Last, we see that EBA is the most robust to small e_t.

6.8 Time shift between GPS and video recorders

We redo the experiments of Section 6.4 for different values of δt, which is the time shift between GPS and video recorders (in seconds). The previous experiences have $\delta t = 0$ and now we try $\delta t \neq 0$ as if the experimenter synchronizes the two recorders manually. Tab. 3 shows that the results are not dramatically corrupted by a bad synchronization: $\delta t \in \{-.5, 0, .5, 1\}$ provides acceptable results, $\delta t > 2$ does not. Such a conclusion depends on the camera speed: the larger speed, the more corrupted GPS location for camera ($\mathbf{x}_1^{gps}$) due to $\delta t \neq 0$, the more difficult fusion. Here the speed is less than 60 km/h.

6.9 Upper bound for track lengths

Tab. 3 shows that the result of SfM-GPS fusion depends on L_∞, the upper bound of the track lengths of image points. The default value is $L_\infty = 5$, e.g. if a point is tracked over 13 consecutive frames, then we split this track into 2 tracks with length 5 and 1 track with length 3. Large L_∞ makes fusion more difficult (except EBA).

6.10 Number of iterations

Tab. 3 shows the fusion results for values of It_{max}. UBA and IBA fail if $It_{max} = 1$. If $It_{max} \in \{2, 3, 4\}$, we see that (1) the location and image errors are acceptable (2) the greater It_{max}, the smaller image error. Furthermore, the greater It_{max}, the smaller α returned by EBA. If $It_{max} = 4$, the mean value is $\bar{\alpha} = 0.0017$. If $It_{max} = 1$, $\bar{\alpha} = 0.13$. Remind that $\alpha > 0$ means that the EBA fusion is incomplete (Section 3.3). Other examples of incomplete EBA fusion are $\bar{\alpha}(\mu = 1.01) = 0.76$ and $\bar{\alpha}(k = 25) = 0.56$.

p	m^{gps}_{uba}	m^{gps}_{iba}	m^{gps}_{eba}	m^{2d}_{uba}	m^{2d}_{iba}	m^{2d}_{eba}
20	36.8	31.9	1.80	1.04	1.04	1.06
30	98.4	33.3	1.96	1.03	1.04	1.07
40	32.3	7.84	9.87	1.03	1.04	1.12
50	153	421	175	1.02	1.03	1.08

Table 5: Location and image errors for incomplete GPS data.

6.11 GPS as a periodic perturbation of SfM

We redo the experiments of Section 6.4 with GPS data such that $\mathbf{l}^i_{gps} = \mathbf{l}^i_{sfm} + 10\begin{pmatrix}\cos(2\pi i/p) & \sin(2\pi i/p) & 0\end{pmatrix}^T$, where p is the period of a circular perturbation around the SfM result and $\begin{pmatrix}0 & 0 & 1\end{pmatrix}^T$ is the vertical direction. Tab. 4 shows the fusion results for several p. For small periods ($p \leq 100$), the fusion is mainly SfM since $m^{sfm}_f < m^{gps}_f$. For large periods ($p \geq 800$), the fusion is mainly GPS since $m^{sfm}_f > m^{gps}_f$. This suggests that the fusion only uses the low frequencies of $i \mapsto \mathbf{l}^i_{gps} - \mathbf{l}^i_{sfm}$.

6.12 Incomplete GPS data

The experiments of Section 6.4 are redone with incomplete GPS data, as if GPS satellites are occluded due to high buildings. $\mathbf{l}^i_{gps}$ is available and fusion is done if and only if (i modulo p) $\in [0, p/2]$, where p is a period. Tab. 5 shows the fusion results for several p. EBA has the best results, but fusion is more difficult for large values of p.

6.13 Experiment for sequence 2

Sequence 2 has the following differences with sequence 1: 3d locations provided by Flytec GPS at 1 Hz, H.264 compressed video at 30 Hz by Gopro camera, 19515 images reduced to 640×480, 4000 keyframes, 5 km long trajectory loop (max speed 77 km/h), the altitude variation is 51 m. The ground truth is unknown. The camera and GPS recorders are manually synchronized.

Fig. 3 shows the recorders, three images of the sequence, and a top view of the GPS and IBA trajectories obtained using the default parameters (Section 6.1). Tab. 6 provides the location and image errors. Now EBA has the best mean of location errors and the image errors are larger than those of sequence 1. RMS e^i_{SfM} ranges from 0.35 to 0.51 pixel (the mean is 0.45), which implies that fusions always have $e^i_f \leq 0.77$ pixel.

7 Conclusion

Two constrained bundle adjustments IBA and EBA were introduced to fuse GPS and Structure-from-Motion data. They enforce an upper bound for the

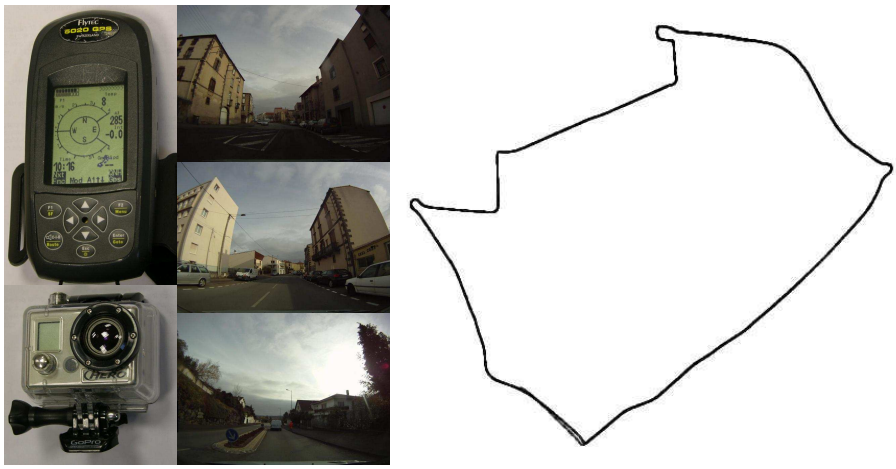

Figure 3: Video (Gopro) and GPS (Flytec) recorders, 3 images of sequence 2, top view of the GPS and IBA trajectories.

f	m_f^{gps}	σ_f^{gps}	∞_f^{gps}	m_f^{2d}	σ_f^{2d}	∞_f^{2d}
SfM	387	224	767	1	0	1
UBA	3.62	3.09	15.1	1.06	0.06	1.43
IBA	2.32	2.84	14.6	1.09	0.05	1.51
EBA	2.12	1.58	10.4	1.13	0.09	1.48

Table 6: Location errors and image errors for sequence 2.

reprojection errors and are described in details. The experiments compare our two BAs with the existing UBA (which minimizes a weighted sum of image and GPS errors) in the difficult context of incremental Structure-from-Motion applied on long urban image sequences and low cost GPS. We also study the fusion results for several parameter settings. Such experiments were not done before.

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). EBA has two advantages: it does not require weight choice and it is the most robust method to bad parameter settings and bad experimental conditions. IBA may provide the best results for good parameter settings. UBA is ranked #3.

Future work includes experiments with other fusion BAs, improvement to initialize the visual reconstruction in the GPS coordinate system, parameter setting from knowledge of the GPS performance, fusion with other sensors, application to georeferenced 3d modeling.

References

[1] S. Agarwal, N. Snavely, S.M. Seitz and R. Szeliski, "Bundle Adjustment in the Large", Proc. ECCV'10.

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

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

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

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

[6] M. Lhuillier, "Fusion of GPS and Structure-from-Motion using constrained Bundle Adjustments", Proc. CVPR'11.

[7] 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", Proc. CVPR'09.

[8] J. McGlone, *The Manual of Photogrammetry*, Fifth ed (ISBN-10 1570830711), ASPRS, 2004.

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

[10] 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, vol. 27, 2009.

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

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

[13] C. Wu, S. Agarwal, B. Curless, and S.M. Seitz, "Multicore Bundle Adjustment", Proc. CVPR'11.