Chapter 1
Incremental Light Bundle Adjustment: Probabilistic Analysis and Application to Robotic Navigation

Vadim Indelman and Frank Dellaert

Abstract   This paper focuses on incremental light bundle adjustment (iLBA), a recently introduced [13] structureless bundle adjustment method, that reduces computational complexity by algebraic elimination of camera-observed 3D points and using incremental smoothing to efficiently optimize only the camera poses. We consider the probability distribution that corresponds to the iLBA cost function, and analyze how well it represents the true density of the camera poses given the image measurements. The latter can be exactly calculated in bundle adjustment (BA) by marginalizing out the 3D points from the joint distribution of camera poses and 3D points. We present a theoretical analysis of the differences in the way that light bundle adjustment and bundle adjustment use measurement information. Using indoor and outdoor datasets we show that the first two moments of the iLBA and the true probability distributions are very similar in practice. Moreover, we present an extension of iLBA to robotic navigation, considering information fusion between high-rate IMU and a monocular camera sensor while avoiding explicit estimation of 3D points. We evaluate the performance of this method in a realistic synthetic aerial scenario and show that iLBA and incremental BA result in comparable navigation state estimation accuracy, while computational time is significantly reduced in the former case.

1.1 Introduction

Bundle adjustment (BA) plays a key role in many applications in mobile vision and robotics. The basic problem can be described as follows: given a sequence of images, determine the maximum a posteriori (MAP) estimate of camera poses and the observed 3D points (or another representation of the observed structure). Fast and

Vadim Indelman and Frank Dellaert

College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA, e-mail: indelman@cc.gatech.edu, dellaert@cc.gatech.edu

reliable calculation of this MAP estimate is important in particular in mobile robotic applications and has been continuously addressed by the research community in recent years.

This paper[1] focuses on the recently developed incremental light bundle adjustment (iLBA) approach [13], that belongs to "structure-less" BA methods [34, 31, 10, 13], in which the camera poses are optimized without including structure parameters into the iterative optimization procedure. iLBA reduces computational complexity by algebraic elimination of 3D points and using an efficient incremental optimization that is based on incremental smoothing (iSAM2) [17, 18].

This paper presents a probabilistic analysis of iLBA, analyzing how well the probability distribution corresponding to the iLBA cost function agrees with the true probability distribution of the camera poses. Accurate and reliable maximum a posteriori and uncertainty estimates are important in many structure from motion and robotic applications, yet to the best of our knowledge this is the *first* time that such an analysis is conducted for structure-less BA methods. This theoretical analysis, which is also valid for other structure-less BA methods, reveals the root effects that cause the iLBA distribution to be an approximation of the true distribution. Using real imagery datasets (see Figures 1.1 and 1.2) we show that in practice the two distributions are close to each other.

In the second part of the paper an extension of iLBA to robotic navigation is presented. We argue that iLBA is in particular suitable for navigation problems, as it facilitates estimation of the navigation state without explicit 3D reconstruction, which is typically not required in the navigation context. Furthermore, it supports loop closure observations (re-observations of 3D points) that are essential for maintaining accurate performance over time. We consider state estimation in the challenging configuration of a monocular camera and high-rate inertial navigation sensors (IMU) and use incremental smoothing to fuse information from these sensors. Similarly to [16], we adopt a recently developed technique [26] for IMU pre-integration to summarize IMU information, significantly reducing the number of variables in the optimization. Finally, we analyze the performance of the proposed method in a synthetic aerial scenario.

Consequently, this paper makes three contributions: 1) probabilistic analysis of iLBA; 2) extension of iLBA to robotic navigation; 3) Evaluation of thereof using real-imagery datasets and a synthetic aerial scenario.

The remainder of this paper is organized as follows. After discussing related work, Section 1.3 overviews the main component of iLBA. Section 1.4 presents a probabilistic analysis of iLBA and evaluation using real-imagery datasets. In Section 1.5 iLBA is extended to robot navigation; this section also includes performance evaluation in a realistic synthetic aerial scenario. Section 1.6 concludes the discussion and suggests directions for future research.

---

[1] Part of the material discussed in this paper was presented in the conference papers [14] and [12].

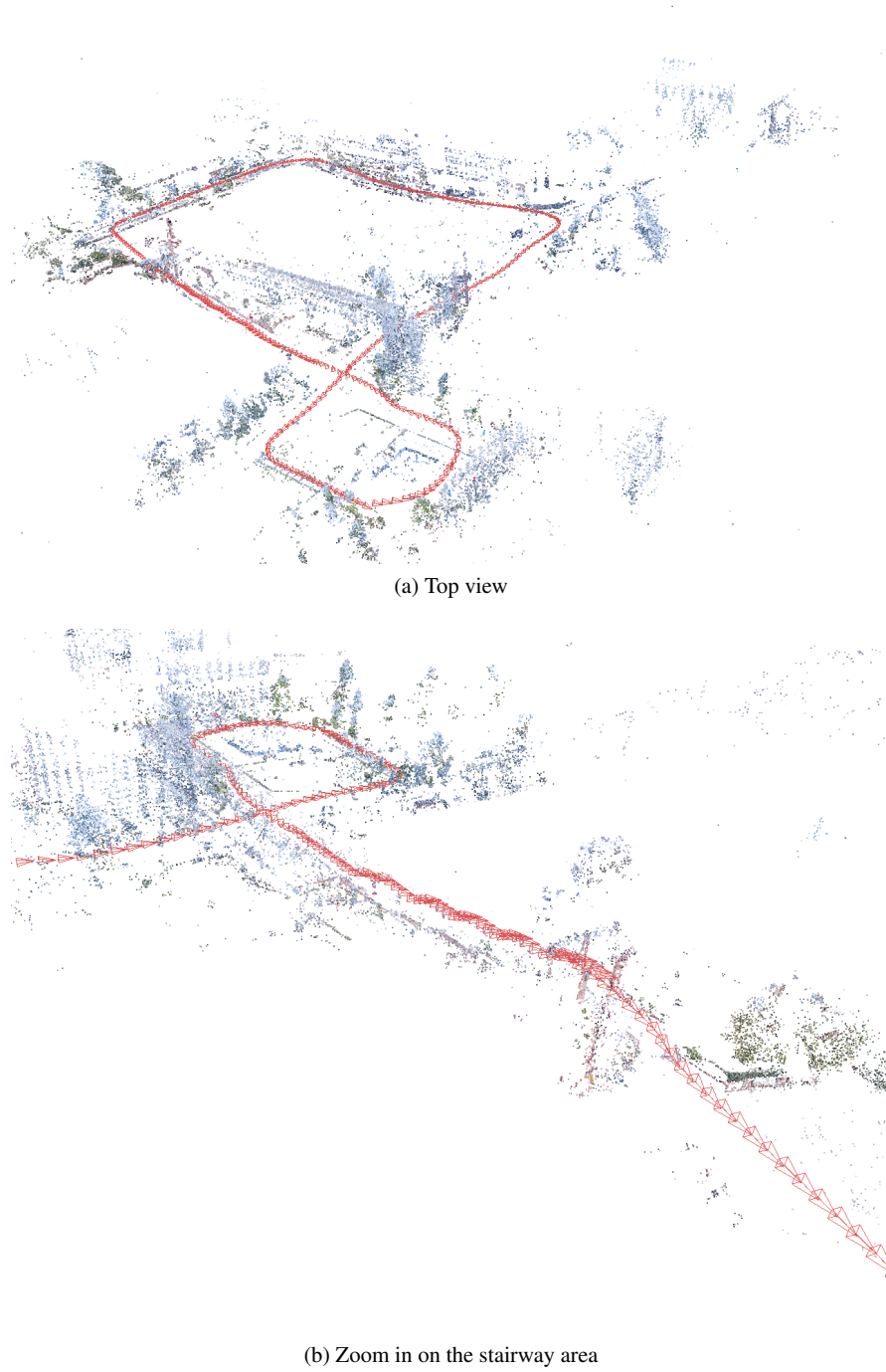(a) Top view

(b) Zoom in on the stairway area

Fig. 1.1: Estimated camera trajectory and sparse 3D reconstruction in the *Outdoor* dataset that is used for evaluating LBA and BA probability distributions.
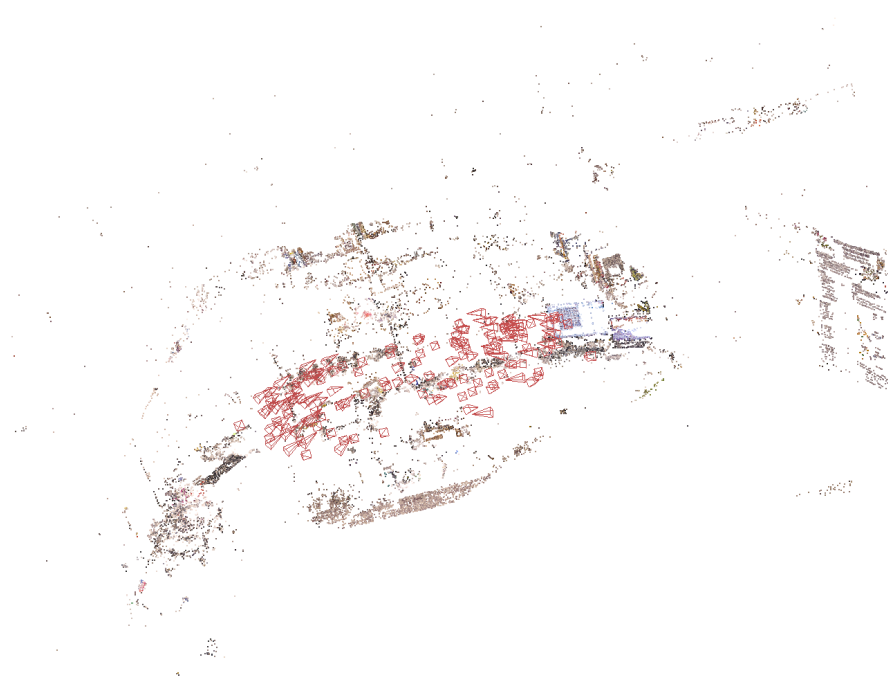
Fig. 1.2: Estimated camera poses and sparse 3D reconstruction in the *Indoor* dataset that is used for evaluating LBA and BA probability distributions.

1.2 Related Work

We organize this section into two parts, discussing first research related to computational efficient bundle adjustment and then reviewing related methods from simultaneous localization and mapping (SLAM) and vision-aided navigation literature.

1.2.1 Computationally Efficient Bundle Adjustment

Development of computationally efficient bundle adjustment approaches, in particular for large-scale scenarios, has become an active research area in the past few years. The developed approaches include methods that exploit sparsity of the involved matrices in the optimization [24, 21], decoupling the BA problem into several submaps that can be efficiently optimized in parallel [30], constructing a skeletal graph using a small subset of images and incorporating the rest of the images using pose estimation [32], solving a reduced version of the non-linear system [22], and

finding a coarse initial solution using a discrete-continuous optimization followed by a BA refinement [2].

Another family of recently suggested methods is structure-less BA [34, 31, 10, 13], in which the camera poses are optimized without including structure parameters into the iterative optimization procedure. The first structure-less BA method was introduced, to the best of our knowledge, by Steffen et al. [34] who optimized the corrections of image observations subject to satisfying trifocal tensor constraints [7]. A similar concept was developed in [10] using three-view constraints [11] instead of the trifocal tensor. Rodrguez et al. [31] obtained reduced computational complexity by reformulating the optimized cost function and refraining from correcting the pixels. Another significant gain in computational complexity was obtained in incremental light bundle adjustment [13], that applied a recently developed technique for incremental smoothing [17, 18] to structure-less BA.

1.2.2 SLAM and Vision-Aided Navigation

Current SLAM algorithms can be divided into two main categories: feature- and view-based SLAM. In feature-based SLAM, both the observed 3D points and the robot's past and current poses are optimized. Several efficient optimization methods that exploit the sparsity of typical structure from motion and SLAM problems have been developed in recent years, some of which are discussed in Section 1.2.1.

The second SLAM category is view-based SLAM [25, 4], or pose-SLAM, in which, similar to iLBA, only the current and past robot's poses are maintained. In pose-SLAM approaches, pairs of poses are linked using relative pose constraints that are straightforward to estimate in a stereo camera setup [8, 22], but become more challenging when relying only on a single camera. In the latter case, the relative constraints can be estimated only up to a scale, which encodes the magnitude of the relative translation [4]. This scale parameter can be set based on the previous frames as in [1]. However, to avoid scale drift the scale parameters should be part of the optimization as well [6]. In contrast to conventional pose-SLAM, iLBA formulates multi-view geometry constraints for *each* feature match, thereby not requiring uncertainty estimates from an intermediate (and separate) process of image-based relative-pose constraints estimation.

In the context of navigation-aiding, despite the close relation to SLAM, only a few methods have been presented in recent years that are capable of incorporating loop closure measurements. These include [28] where visual observations are incorporated into the navigation solution using an EKF formulation with a sliding window of past poses. In a later work [29], the authors applied a conventional batch BA that involved explicit structure estimation in order to handle loop closure measurements. More recently, incremental smoothing [18] was proposed for inertial navigation systems in [15, 16] and a method was developed to incorporate loop closures while maintaining a real time navigation solution [20]. The extension of iLBA to robotic navigation that is described in this paper is formulated within the same

framework of [20, 15, 16] but replaces the explicit estimation of 3D points with a set of 2-view and 3-view constraints.

1.3 Incremental Light Bundle Adjustment

Incremental light bundle adjustment (iLBA) [13] combines the following two key-ideas: algebraic elimination of 3D points, and incremental smoothing. In this section we review each of these concepts, first introducing notations and standard bundle adjustment formulation.

1.3.1 Bundle Adjustment

Consider a sequence of N views observing M 3D points, and denote the ith camera pose by x_i and the measured image observation of the jth 3D point l_j by z_i^j. Let also $X \doteq \left\{ x_1^T, \ldots, x_N^T \right\}^T$ and $L \doteq \left\{ l_1^T, \ldots, l_M^T \right\}^T$.

The joint pdf $p(X, L|Z)$ can be explicitly written in terms of the prior information and the actual measurement models:

$$p(X, L|Z) = priors \cdot \prod_i \prod_{j \in \mathcal{M}_i} p\left(z_i^j | x_i, l_j \right), \tag{1.1}$$

where $p\left(z_i^j | x_i, l_j \right)$ is the measurement model corresponding to the probability density of observing the 3D point l_j from a camera pose x_i at the pixel location z_i^j, and $\mathcal{M}_i$ represents the indices of all 3D points observed from the ith camera. Assuming Gaussian distributions, the maximum a posteriori (MAP) estimation

$$X^*, L^* = \arg\max_{X,L} p(X, L|Z),$$

corresponds to minimizing the following nonlinear cost function (omitting prior terms for clarity)

$$J_{BA}(X, L) = \sum_i \sum_{j \in \mathcal{M}_i} \left\| z_i^j - \pi(x_i, l_j) \right\|_{\Sigma}^2, \tag{1.2}$$

where $\pi(\cdot)$ is the projection function [7] for a standard pinhole camera model, and $\|a\|_{\Sigma}^2 \doteq a^T \Sigma^{-1} a$ is the squared Mahalanobis distance with the measurement covariance matrix Σ. Each term in the cost function J_{BA} corresponds to the re-projection error between the measured and predicted image observations (see Figure 1.3).
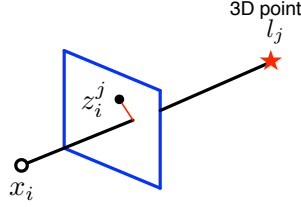
Fig. 1.3: Illustration of re-projection error: difference between projection of the 3D point l_j onto the camera frame using camera pose x_i, and the image observation z_i^j.

1.3.2 Algebraic Elimination of 3D points using Three-View Constraints

Performing inference over the joint pdf $p(X, L|Z)$ involves optimizing $6N + 3M$ variables, with N and M denoting camera frames and observed 3D points, respectively. Instead, in this section we reduce the number of variables to only $6N$ by algebraically eliminating all the 3D points.

Considering all the camera frames that observe some 3D point l_j and writing down all the appropriate projection equations, it is possible to algebraically eliminate l_j, leading to multiple view constraints that involve up to triplets of cameras [27, 35]. One possible formulation of these constraints, recently developed in the context of vision-aided navigation [9, 11], is the three-view constraints. The close relation between these constraints and the well-known trifocal tensor was reported in [13]. The three view constraints, for a triplet of camera frames k, l and m are given by (see Figure 1.4)

$$g_{2v}\left(x_k, x_l, z_k^j, z_l^j\right) = q_k \cdot (t_{k \to l} \times q_l) \tag{1.3}$$

$$g_{2v}\left(x_l, x_m, z_l^j, z_m^j\right) = q_l \cdot (t_{l \to m} \times q_m) \tag{1.4}$$

$$g_{3v}\left(x_k, x_l, x_m, z_k^j, z_l^j, z_m^j\right) = (q_l \times q_k) \cdot (q_m \times t_{l \to m}) - (q_k \times t_{k \to l}) \cdot (q_m \times q_l) \tag{1.5}$$

where q_i is the line of sight expressed in a global frame, $q_i \doteq R_i^T K_i^{-1} z$, for any view i and image observation z, K_i is the calibration matrix of this view, R_i represents the rotation matrix from the global frame to the ith view's frame, and $t_{i \to j}$ denotes the translation vector from view i to view j, expressed in the global frame. The first two constraints are the two-view constraints g_{2v} between appropriate pairs of views, also known as the epipolar constraints. Given matching image observations, these constraints allow to recover the relative rotation and up-to-scale translation motion between camera pairs. The third constraint, g_{3v}, involves all the three views and enforces a consistent scale of the translation vectors $t_{k \to l}$ and $t_{l \to m}$. Thus, if the magnitude of the former is known, the magnitude of the later can be determined.
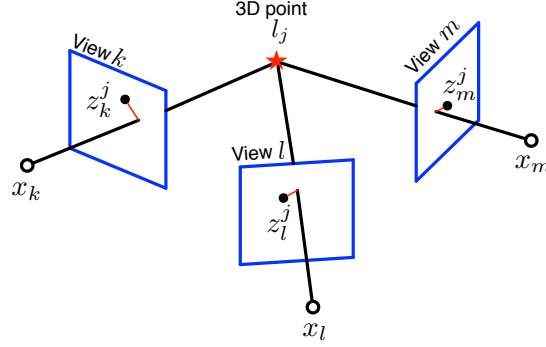
Fig. 1.4: Three camera frames k, l and m observing the same 3D point l_j. Explicit estimation of l_j can be avoided by algebraic elimination using three-view constraints (1.3)-(1.5).

When a 3D point is observed by more than three views, a single two-view and three-view constraint between each new view m and past views k, l is added [10]. Determining which past views to use is still an open question that we currently investigate. In the results reported in this paper, we heuristically choose the past views to be the earliest camera frame k observing the 3D point and set l to be the middle camera frame such that the translation vectors $t_{k \to l}$ and $t_{l \to m}$ are of similar magnitudes[2].

In practice, in order to avoid the trivial solution of zero translation, we normalize each of the constraints g_{2v} and g_{3v} by a translation vector and modify the Jacobian matrices accordingly.

Algebraically eliminating all the 3D points using the three-view constraints (1.3)-(1.5) leads to the following cost function that is expressed in terms of these constraints, instead of re-projection errors as in bundle adjustment (Eq. (1.2)):

$$J_{LBA}(X) \doteq \sum_{i=1}^{N_h} \| h_i(X_i, Z_i) \|_{\Sigma_i}^2, \tag{1.6}$$

where $h_i \in \{g_{2v}, g_{3v}\}$ represents a single two- or three-view constraint that is a function of several camera poses $X_i \subset X$ and image observations Z_i in the appropriate views, and N_h is the overall number of such constraints. One can observe that the cost function J_{LBA} does not contain any 3D points as variables.

Note that an exact marginalization of 3D points $p(X|Z) = \int_L p(X, L|Z)\, dL$ is another alternative, however, as discussed in the sequel, it densifies the matrices and therefore will not necessarily result in computational gain. Algebraic elimination of 3D points following the approach discussed above avoids some of this densifi-

---

[2] When adding a new camera into the optimization we initialize its pose using a three-view constraint g_{3v}, while keeping the poses of the other two cameras fixed.

cation by discarding a certain amount of information. As a result, the inference is performed only over the camera poses and is typically significantly faster than conventional bundle adjustment at the cost of a certain degradation in accuracy. The analysis of this tradeoff is further discussed in Section 1.4.

1.3.3 Incremental Smoothing

The second component in iLBA is the recently-developed incremental smoothing [19, 18], an efficient nonlinear optimization technique that exploits sparsity and reuses calculations when possible. Below we review the main concepts of this technique, and refer the reader to [19, 18] for further details.

1.3.3.1 Factor Graph Representation

The incremental smoothing technique uses the factor graph graphical model [23] to represent a given factorization of the joint probability distribution function (pdf). Formally, a factor graph is a bipartite graph $G = (\mathscr{X}, \mathscr{F}, \mathscr{E})$ with $\mathscr{X}, \mathscr{F}$ being variable and factor nodes and $\mathscr{E}$ consisting of edges that connect between these two variable groups. Each ith probabilistic term $p(.)$ in the factorization of the joint pdf is represented by a *factor* node $f \in \mathscr{F}$ that is connected by edges $e \in \mathscr{E}$ to *variable* nodes $\mathscr{X}_i \subset \mathscr{X}$ that are involved in $p(.)$.

In case of bundle adjustment, the factorization of the joint pdf $p(X, L|Z)$ is given by Eq.(1.1), and defining the projection factor for some view x, landmark l and image observation z as

$$f_{proj}(x,l) \doteq \exp\left(-\frac{1}{2}\|z - proj(x,l)\|_{\Sigma}^2\right),$$

the factor graph formulation can be trivially written as

$$p(X, L|Z) \propto \prod_i \prod_{j \in \mathscr{M}_i} f_{proj}(x_i, l_j).$$

In order to represent LBA in a factor graph, a suitable joint probability distribution function $p_{LBA}(X|Z)$ should be formulated first. Since the residual errors of three-view constraints (1.3)-(1.5) have been shown [13] to be of Gaussian distribution, the LBA pdf and the two- and three-view factors can be defined as

$$p_{LBA}(X|Z) \propto \prod_{i=1}^{N_h} f_{2v/3v}(X_i, Z_i), \tag{1.7}$$

and

$$f_{2v}(x_k, x_l) \doteq \exp\left(-\frac{1}{2}\|g_{2v}(x_k, x_l, z_k, z_l)\|^2_{\Sigma_{2v}}\right) \qquad (1.8)$$

$$f_{3v}(x_k, x_l, x_m) \doteq \exp\left(-\frac{1}{2}\|g_{3v}(x_k, x_l, x_m, z_k, z_l, z_m)\|^2_{\Sigma_{3v}}\right), \qquad (1.9)$$

where the covariance matrices Σ_{2v} and Σ_{3v} are given in [13].

Figure 1.5b illustrate factor graphs that represent $p(X, L|Z)$ and $p_{LBA}(X|Z)$ in a simple case of 4 camera poses observing 2 different 3D points. Each method uses different factors as discussed above.

1.3.3.2 Incremental Inference

Computing a MAP estimate of a given joint pdf typically involves a nonlinear optimization, where in each iteration a linearized system of the form $\Delta^* = \arg\min_\Delta (A\Delta - b)$ is solved. To that end, the *large sparse* Jacobian matrix A is factorized, e.g. using QR factorization, and an equivalent system $R\Delta - d$, with $d \doteq Q^T b$ is obtained. Since R is upper triangular, this system can be easily solved, in a process known as back-substitution.

Instead of calculating a factorization of the Jacobian A from scratch, the matrix R from the previous factorization can be updated with new information. Incremental smoothing performs this operation very efficiently using graphical models [19, 18]: The factor graph is eliminated into a Bayes net using a calculated elimination order, and can be also converted into Bayes tree. Both graphical models represent the sparse factorized matrix R (which is the square root information matrix). Updating a factorization involves identifying what parts in the Bayes net (tree) are affected and re-eliminating only these variables. Additionally, tracking the validity of linearization point of each variable allows to perform selective re-linearization, instead of always re-linearizing all variables, while still recovering the MAP estimate up to a tolerance [19, 18].

1.4 Probabilistic Analysis of Light Bundle Adjustment

This section analyzes how well the LBA distribution $p_{LBA}(X|Z)$ represents the true density $p(X|Z)$. An exact calculation of the latter would marginalize the landmarks from the joint $p(X, L|Z)$

$$p(X|Z) = \int_L p(X, L|Z)\, dL.$$

While in practice, LBA represents a similar probability density over cameras as BA, there are two root effects that cause the LBA distribution to be an approximation of the true density: First, *LBA discards some mutual information in large*
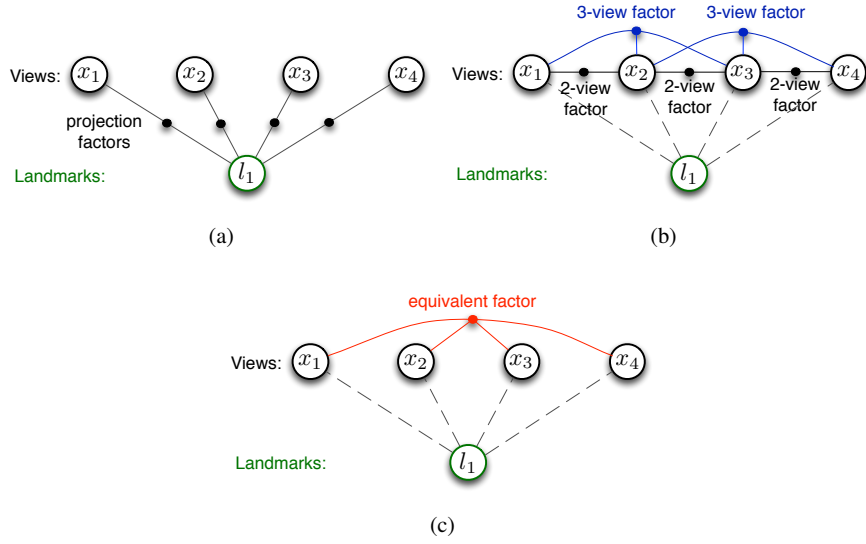
Fig. 1.5: Factor graph formulation for (a) BA and (b) LBA. (c) Factor graph after marginalizing out the landmark l_1.

camera cliques, by considering only the mutual information between camera pairs and triplets introduced by them observing the same landmark. Bundle adjustment, on the other hand, induces mutual information between *all* cameras observing the same landmark. Second, *LBA duplicates some information for image measurements used in multiple factors*, double-counting measurements that appear in multiple two- or three-view factors.

As an example of both of these effects, consider observing a landmark l by four views x_1, x_2, x_3 and x_4, as illustrated in Figure 1.5. The joint pdf is given by

$$p(X_4, l | Z_4) \propto \prod_{i=1}^{4} f_{proj}(x_i, l), \tag{1.10}$$

where X_4 and Z_4 denote the four camera poses and the four image observations, respectively. On the other hand, the LBA pdf is

$$p_{LBA}(X_4 | Z_4) \propto f_{2v}(x_1, x_2) f_{2v}(x_2, x_3) f_{3v}(x_1, x_2, x_3)$$
$$f_{2v}(x_3, x_4) f_{3v}(x_2, x_3, x_4) \tag{1.11}$$

which corresponds to the set of two- and three-view factors, as shown in Figure 1.5.

The first effect, discarding of mutual information, can be seen when comparing the LBA pdf with the pdf resulting from eliminating the landmarks from the BA pdf,

$$p(X_4|Z_4) = \int_{X_4} p(X,L|Z)\,dX_4 = p(x_1,x_2,x_3,x_4|z_1,z_2,z_3,z_4) \qquad (1.12)$$

The result in the case of BA is a single clique over all cameras. In general, there is no way to exactly factor such a dense clique in a way that reduces complexity. The multiple factors of LBA over pairs and triplets (Eq. (1.11)) reduce complexity instead by discarding some "links" that would otherwise be introduced between cameras.

The second effect, duplication of some image measurement information, can be seen in the sharing of cameras between LBA factors in Eq. (1.11). Any two factors sharing a camera in common both use the information from the shared camera, effectively duplicating it. For example, $f_{2v}(x_1,x_2)$ and $f_{2v}(x_2,x_3)$ both use the information from the measurements in camera 2.

This duplication of information happens since the two- and three-view factors were assumed to have independent noise models, represented by the covariance matrix Σ_i for the ith factor, and therefore could be separately written in the LBA pdf (1.7). This assumption is violated for factors that share measurements, as in the example above. One approach to avoid double counting is therefore to augment such factors, while accounting for the fact that the same measurement is involved by appropriate cross-covariance terms in the (augmented) covariance matrix. For example, for any two factors representing constraints g_a and g_b, we can define $f_{aug} \doteq \exp\left(-\frac{1}{2}\|g_{aug}\|^2_{\Sigma_{aug}}\right)$, with $g_{aug} \doteq \begin{bmatrix} g_a & g_b \end{bmatrix}^T$ and the augmented covariance matrix Σ_{aug}:

$$\Sigma_{aug} \doteq \begin{bmatrix} \Sigma_a & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_b \end{bmatrix} \qquad (1.13)$$

where the cross-covariance terms Σ_{ab} are non-zero when the constraints share measurements. However, since multiple factors are combined into a single multi-dimensional factor that involves all the variables in these individual factors, the factor graph becomes denser. Therefore, such an approach is expected to have considerable impact on computational complexity.

As we show in the next section, despite the above two aspects, the actual LBA distribution is very similar to the true distribution $p(X|Z)$. It is worth mentioning that the presented probabilistic analysis is valid for other existing structure-less BA methods [34, 31, 10] as well.

1.4.1 Datasets for Evaluation and Implementation

We use two datasets to evaluate how well the iLBA distribution $p_{LBA}(X|Z)$ represents the true density $p(X|Z)$. In the first dataset (*Cubicle*) the camera observes a cubicle desk in an open space environment from different viewpoints and distances (see Figure 1.2). In the second dataset, *Outdoor*, the camera follows a trajectory encircling a courtyard and building and performing loop closures as shown in Figures

1.1 and 1.6. Figure 1.7 shows typical images from these two datasets, while Table 1.1 provides further details regarding the number of views (N) and 3D points (M), as well as the number of total observations in the two datasets.
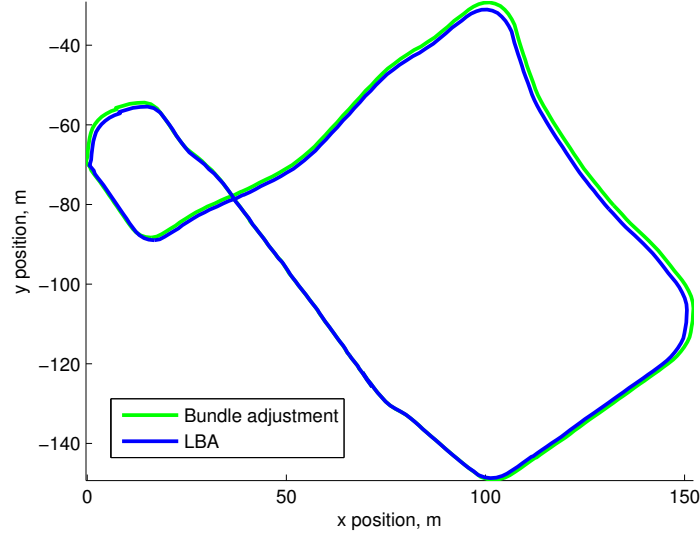

Fig. 1.6: Estimated trajectory in *Outdoor* dataset. LBA and conventional BA produce very similar results.

Dataset	N, M, #Obsrv	Avg. reproj. error [pix]		Overall time [s]	
		iLBA	iBA	iLBA	iBA
Cubicle	148, 31910, 164358	0.552 pix	0.533 pix	581	6024
Outdoor	308, 74070, 316696	0.418 pix	0.405 pix	3163	26414

Table 1.1: Dataset details and performance of iLBA and BA: Re-projection errors and computational cost using incremental smoothing in all methods.

All methods were implemented using the GTSAM factor graph optimization library[3] [3, 18]. Incremental smoothing was used in all cases, denoted by the prefix *i* (i.e. iLBA and iBA). Image correspondences, as well as the camera calibration matrices, were obtained by first running Bundler[4] [33] on each dataset. Additional implementation details can be found in [13].

[3] https://borg.cc.gatech.edu/.

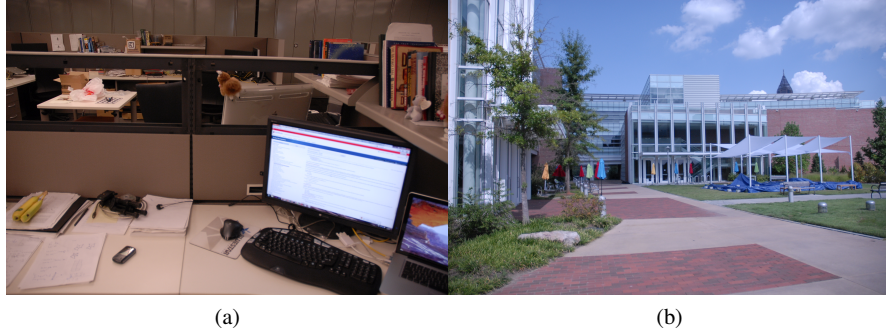[4] http://phototour.cs.washington.edu/bundler.

Fig. 1.7: Typical images in the *Cubicle (a)* and *Outdoor* (b) datasets.

1.4.2 Evaluation

In this section we compare the distributions of iLBA and incremental BA using two real imagery datasets. We first discuss how this comparison is made, present MAP estimate and computational cost of each method in Section 1.4.2.2 and then focus on estimated uncertainties by the two approaches in Section 1.4.2.3.

1.4.2.1 Method for Comparing the PDFs of LBA and BA

Because computing the true marginal over cameras for BA $p(X|Z)$ is not tractable in closed form, we use an alternate method to compare the pdfs of LBA and BA. This method evaluates how well LBA and BA agree in both the absolute uncertainty of each camera in a global frame, and the relative uncertainty between all pairs of cameras.

In order to compare uncertainties, we first assume that $p_{LBA}(X|Z)$ and $p(X|Z)$ both are well-approximated as multivariate Gaussian distributions about their MAP estimates

$$p_{LBA}(X|Z) = N(\mu_{LBA}, \Sigma_{LBA})$$

$$p(X|Z) = N(\mu, \Sigma).$$

In order to compare relative uncertainty between cameras, we compare conditional densities $p(x_i|x_j, Z)$ between all pairs of cameras. This calculation quantifies how well LBA agrees with BA in relative uncertainty, while avoiding calculating the full covariance matrix on all cameras, which quickly becomes intractable for large numbers of cameras. The conditionals are obtained by integrating out all variables other than x_i and x_j,

$$p\left(x_i|x_j,Z\right) = \int\limits_{X\setminus\{x_i,x_j\},L} p\left(X,L|Z\right)/p\left(x_j|Z\right).$$

In practice, we do this analytically by approximating the joint as a Gaussian around its MAP estimate, and applying sparse factorization,

$$p\left(X,L|Z\right) = p\left(X\setminus\{x_i,x_j\},L|x_i,x_j,Z\right)p\left(x_i|x_j,Z\right)p\left(x_j|Z\right) \tag{1.14}$$

from which the desired conditional $p\left(x_i|x_j,Z\right)$ can be read off.

1.4.2.2 MAP estimate and Computational Cost

Before discussing probabilistic aspects, we show performance results, in terms of accuracy of the MAP estimate and computational complexity. As seen in Table 1.1 and Figure 1.6, while iLBA yields a similar, but a bit degraded accuracy, the computational cost of iLBA is *8-10 times faster* than incremental BA.

### 1.4.2.3 Estimated Camera Pose Uncertainty

We compare the probability density of the cameras estimated by iLBA to that of incremental BA by comparing their discrepancy both in the marginal uncertainty of each camera, and in relative uncertainty between each camera pair, as described in Section 1.4.2.1. We provide details as to how this comparison was made in the Appendix.

A comparison of the absolute uncertainty for the *Cubicle* dataset is given in Figure 1.8 and Figures 1.9a-1.9b. Figure 1.8a compares, for each camera pose i, between the covariance trace of Σ^i_{LBA} and Σ^i_{BA}. As seen, the initial uncertainty is very small and it increases as the camera moves around the cubicle deck and drops to low values when the camera captures previously-observed areas thereby providing loop-closure measurements. Figure 1.8b describes the interaction between the uncertainty of each view and the number of factors that involve this view. As expected, it can be seen that the covariance is higher when less factors are involved and vice versa.

Overall, the absolute uncertainties in LBA and BA are very similar. This can be also observed in Figures 1.9a-1.9b that show a histogram of the discrepancy (1.19) both for position and orientation terms. Typical position discrepancies are near -10^{-4} meters. The discrepancies for relative uncertainties are given in Figures 1.9c-1.9d for position and orientation terms.

Figure 1.10 shows the discrepancy histograms for the *Outdoor* dataset. The absolute and relative discrepancies between LBA and BA are small, e.g. less than 5 centimeters in the absolute position for a trajectory that spans an area of 120×150 meters (cf. Figure 1.6), and on the order of 10^{-4} radians for the absolute rotation uncertainty.

Our conclusion from this evaluation is that the uncertainties estimated using iLBA represent well the uncertainties of incremental BA and therefore can be used instead of the later, e.g. in the context of establishing data association.
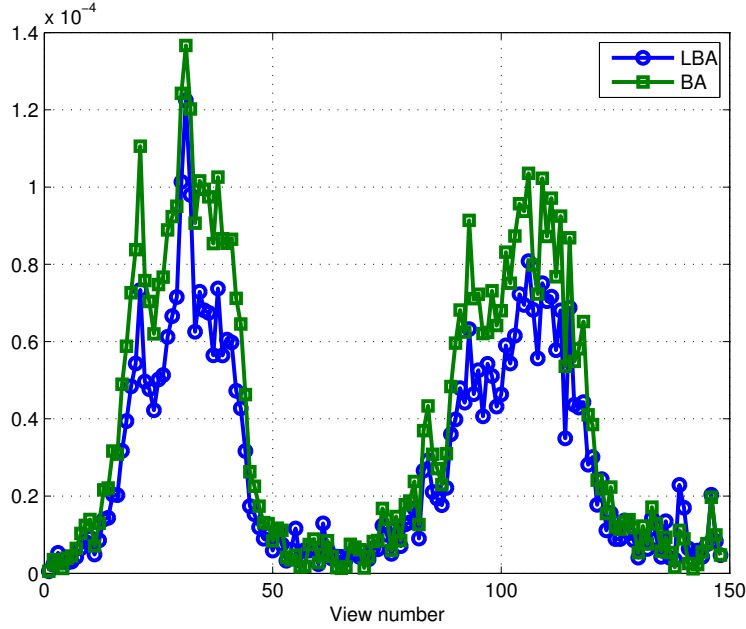
(a)
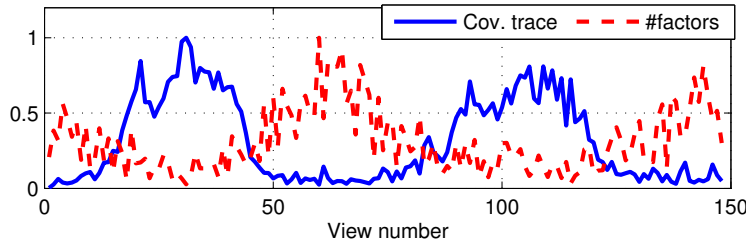
(b)

Fig. 1.8: *Cubicle* dataset: (a) Covariance trace of each camera pose. (b) Trace of covariance and number of factors in LBA formulation, both are normalized to 1.
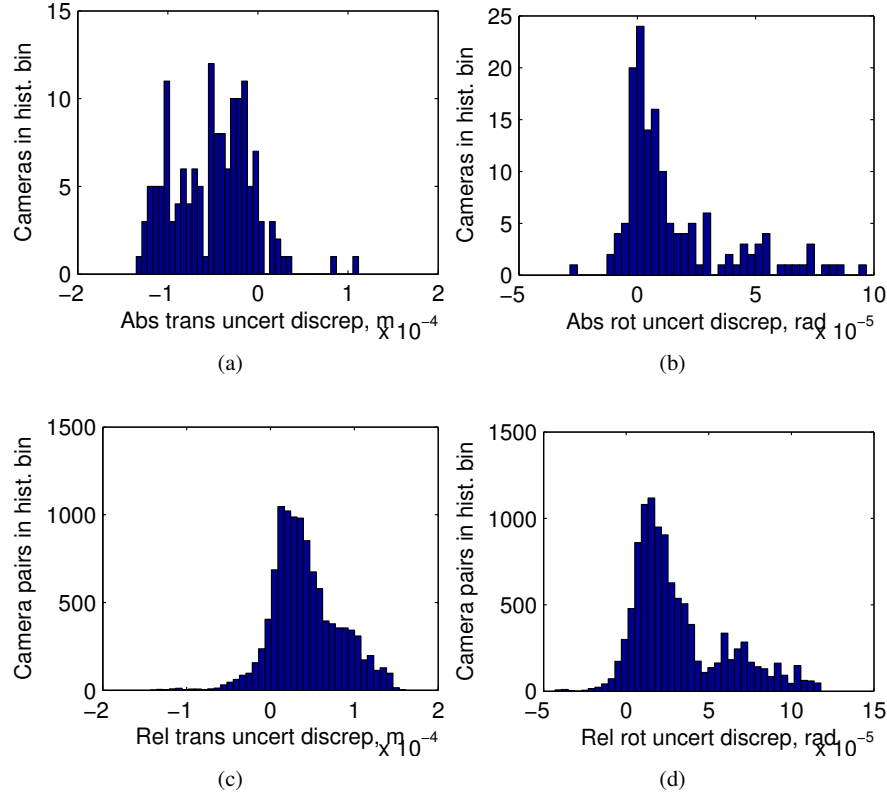
Fig. 1.9: Discrepancy histograms for the *Cubicle* dataset: Absolute position (a) and orientation (b); Relative position (c) and orientation (d) between every camera pair in the sequence.

1.5 Application iLBA to Robotic Navigation

While incremental light bundle adjustment has been discussed thus far in the context of structure from motion problems, it is particularly attractive also to robotic navigation where different sensors are often available.

In this section we extend iLBA to robotic navigation and consider the challenging configuration of a robot equipped only with high-rate inertial navigation sensors (IMU) and a monocular camera. We show that this information fusion problem can be solved using incremental smoothing and adapt a recently-developed technique [26] for summarizing consecutive IMU measurements to obtain high-rate performance. We present proof-of-concept results using a synthetic aerial scenario.

Slightly abusing the previous notation, we redefine x to be the navigation state, comprising robot pose (position and orientation) and velocity. The IMU calibration
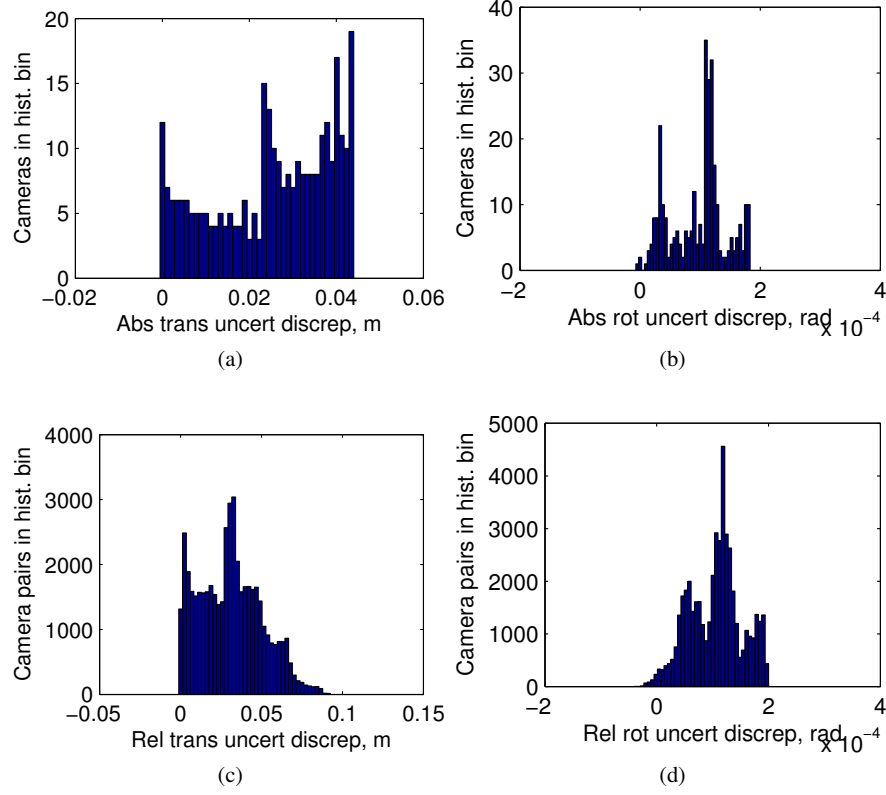
Fig. 1.10: Discrepancy histograms for the *Outdoor* dataset: Absolute position (a) and orientation (b); Relative position (c) and orientation (d) between every camera pair in the sequence.

parameters are denoted by b; although in this section no specific parametrization is assumed, we will refer to b as IMU bias.

1.5.1 Formulation

As standard in navigation literature [5], we use the probabilistic motion model

$$p\left(x_{k+1}|x_k,b_k,z_k^{IMU}\right) \propto \exp\left(-\frac{1}{2}\left\|x_{k+1}-h^{IMU}\left(x_k,b_k,z_k^{IMU}\right)\right\|_{\Sigma_{IMU}}^2\right) \doteq f^{IMU}\left(x_{k+1},x_k,b_k\right)$$

$$(1.15)$$

to represent the distribution over the state x_{k+1} given previous state x_k, an IMU measurement z_k^{IMU} and IMU calibration, that we will refer to as bias, b_k. The function h^{IMU} represents the nonlinear discrete inertial navigation equations [5]. The time evolution of IMU bias is modeled using some dynamics function h^b and is expressed probabilistically as

$$p(b_{k+1}|b_k) \propto \exp\left(-\frac{1}{2}\left\|b_{k+1} - h^b(b_k)\right\|_{\Sigma_b}^2\right) \doteq f^{bias}(b_{k+1}, b_k). \qquad (1.16)$$

Fusing information from IMU and camera sensors using LBA framework then involves calculating the MAP estimate of the following joint probability distribution function:

$$p(X_k, B_k|Z_k) \propto \prod_{s=0}^{k-1}\left[f^{IMU}(x_{s+1}, x_s, b_s)f^{bias}(b_{s+1}, b_s)\prod_{i=1}^{n_s} f_{2v/3v}(X_{s_i})\right], \qquad (1.17)$$

where we used the IMU and bias factors f^{IMU} and f^{bias} defined in Eqs. (1.15)-(1.16), with Σ_{IMU} and Σ_b representing the corresponding process covariance matrices, and the overall set of IMU biases denoted by $B_k \doteq \left[b_1^T \cdots b_k^T\right]^T$; in practice, since these tend to only have slow dynamics, it makes sense to describe this process in some lower rate [15, 16]. In Eq (1.17), n_{s+1} is the number of two- and three-view factors that are added between each current state x_{s+1} and past states. Thus, if $x_a \in X_{s_i}$ then $a \leq s+1$.

While the MAP estimate $X_k^*, B_k^* = \arg\max_{X_k, B_k} p(X_k, B_k|Z_k)$ can be calculated using incremental smoothing, high-rate performance becomes infeasible: Number of variables in the optimization rapidly increases as a new navigation state is introduced at IMU rate (for each new IMU factor). Moreover, number of variables that need to be re-calculated rapidly increases when new two- and three-view factors are added to the graph.

To get a better understanding of this aspect, it is beneficial to first consider only IMU observations. Adding new IMU and bias factors involve only re-eliminating the two past navigation and bias states *regardless* to the graph size. This is illustrated in Figure 1.11 for two consecutive time instances t_4 and t_5. The figure shows both factor graphs and bayes nets with the latter representing the square root information matrix R. The nodes that were modified from the previous Bayes net are shown in Figure 1.11d in red.

Considering now also a camera sensor, adding two- and three-view factors (or projection factors for bundle adjustment) would require re-eliminating many more variables. While the exact number depends on variable elimination ordering, typically at least the variables in between the variables involved in the new factors will have to be re-eliminated. For example, adding a single two-view factor (Figure 1.12a) most probably will involve re-eliminating the majority of the variables $x_1 - x_5$ and $b_1 - b_4$. Adding other multi-view factors that involve additional nav-

igation states will require re-eliminating many more variables, and thus high-rate performance is only possible for a limited time.

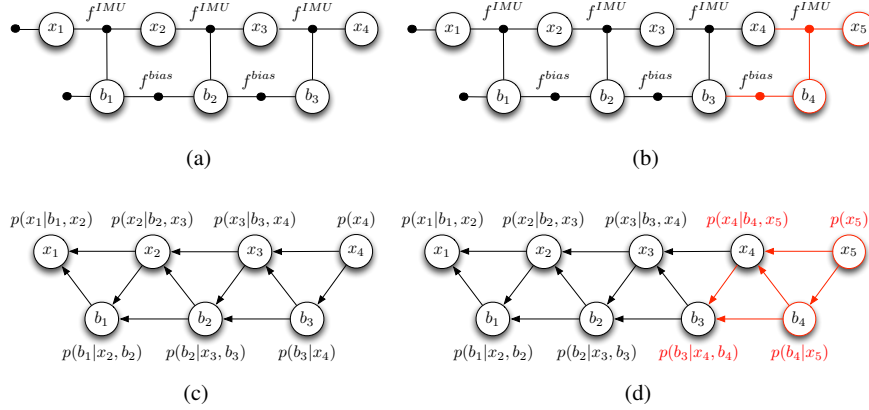In the next section we discuss a solution to this problem.

Fig. 1.11: (a)-(b) Factor graphs and (c)-(d) Bayes nets for the pure-IMU case in two consecutive time instances. Adding IMU and bias factors involves re-eliminating only 2 past nodes. Modified parts are marked in red.

1.5.2 Equivalent IMU Factor

In this section we adopt a recently-developed technique [26] for IMU measurements pre-integration that allows to reduce the number of variables and factors in the optimization, resulting in significantly improved computational complexity.

The idea is to integrate consecutive IMU measurements between two time instances t_i and t_j into a single component, denoted by $\Delta x_{i \to j}$, comprising the accumulated change in position, velocity and orientation, represented respectively by $\Delta p_{i \to j}, \Delta v_{i \to j}$ and the rotation matrix R^i_j:

$$\Delta x_{i \to j} \doteq \left\{ \Delta p_{i \to j}, \Delta v_{i \to j}, R^i_j \right\} = \eta \left(Z^{IMU}_{i \to j}, b_i \right),$$

where $Z^{IMU}_{i \to j}$ is the set of IMU measurements between the time instances t_i and t_j, that are corrected using the bias b_i, and η is a known non-linear function that describes the IMU measurements pre-integration process. One can now use $\Delta x_{i \to j}$ to predict x_j based on the current estimate of x_i. Let h^{Equiv} represent this predicting function.

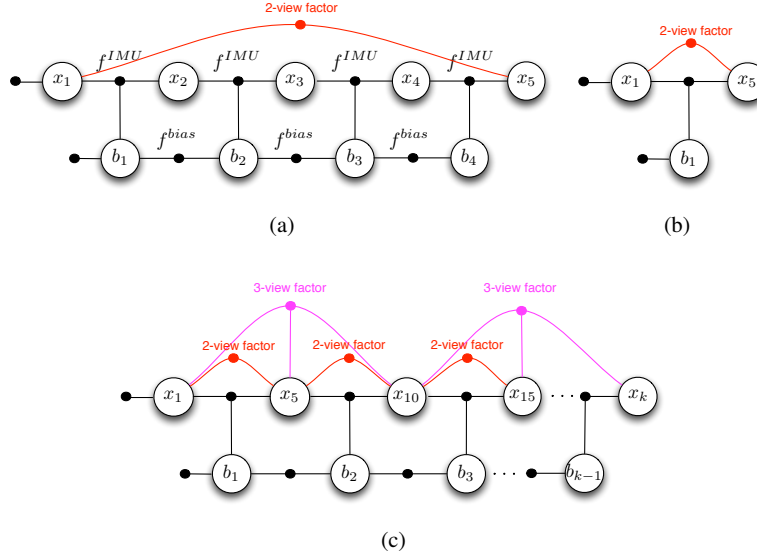We can then define an *equivalent* IMU factor [16] f^{Equiv} as

Fig. 1.12: (a) Factor graph with a single two-view factor with IMU and bias factors; (b) The corresponding factor graph using an equivalent IMU factor; (c) Factor graph with many two- and three-view factors as well as equivalent IMU and bias factors.

$$f^{Equiv}(x_j, x_i, b_i) \doteq \exp\left(-\frac{1}{2}\left\|x_j - h^{Equiv}(x_i, b_i, \Delta x_{i\to j})\right\|_{\Sigma}^2\right), \qquad (1.18)$$

which involves only the variables x_j, x_i and b_i for any reasonable[5] two time instances t_i and t_j. Figure 1.12b illustrates the conceptual difference between the conventional and equivalent IMU factors.

The approach for calculating $\Delta x_{i\to j}$ involves pre-integrating the IMU measurements while expressing them in the navigation frame. However, this will require re-calculating $\Delta x_{i\to j}$ from scratch each time the rotation estimate changes, i.e. each re-linearization of x_i. To resolve this, as proposed in [26], the different components in $\Delta x_{i\to j}$ are expressed in the body frame of the first time instant (i.e. t_i), which allows re-linearizing the factor (1.18) without recalculating $\Delta x_{i\to j}$. The reader is referred to [26] for further details.

The equivalent IMU factor allows to significantly reduce the number of variables and factors in the optimization, and enables high-rate performance while using efficient optimization techniques. This is illustrated in Figure 1.12c, that shows a factor graph with two- and three-view factors and the equivalent IMU factor bridg-

---

[5] The original derivation in [26] neglects Earth curvature and Earth rotation, however it can be extended to the more general case which assumes the gravity vector and the rotation rate of the navigation frame with respect to an inertial frame are constant. The time instances t_i, t_j should be chosen such that these assumptions are satisfied.

ing between navigation states (variables) from different time instances. Note that a conventional IMU factor would require adding consecutive navigation states to the graph.

Furthermore, since in typical navigation systems a navigation solution x_t is required in real time, i.e. each time an IMU measurement is obtained, one can predict x_t using the accumulated component $\Delta x_{i \to j}$ and the current estimates $\hat{x}_i, \hat{b}_i$ of x_i and b_i, in a parallel process and *without* incorporating x_t into the optimization, i.e. $h^{Equiv}\left(\hat{x}_i, \hat{b}_i, \Delta x_{i \to j}\right)$.

1.5.3 Evaluation in a Simulated Aerial Scenario

In this section we present an evaluation of the described extension of LBA to robotic navigation in a realistic aerial scenario covering an area of about 2×1.5 km as shown in Figure 1.13a. We also compare LBA to the BA approach, both using the equivalent IMU factors and incremental smoothing. A statistical study of the approach using a smaller scenario is reported in a conference version of this paper [12].

In the simulated scenario, the aerial vehicle gradually explores different areas and occasionally re-visits previously observed locations thereby providing loop closure measurements. The flight is at a constant height of 200 meter above mean ground level, with a 40 m/s velocity. The aerial vehicle travels a total distance of about 13 km in 700 seconds. A medium-grade IMU and a single downward-facing camera, operating at 100 Hz and 0.5 Hz, were used.

The 100 Hz ideal IMU measurements were corrupted with a constant bias and a zero-mean Gaussian noise in each axis. Bias terms were drawn from a zero-mean Gaussian distribution with a standard deviation of $\sigma = 10$ mg for the accelerometers and $\sigma = 10$ deg/hr for the gyroscopes. The noise terms were drawn from a zero-mean Gaussian distribution with $\sigma = 100 \, \mu g / \sqrt{Hz}$ and $\sigma = 0.001 \, deg / \sqrt{hr}$ for the accelerometers and gyroscopes. Visual observations of unknown 3D points were corrupted by a zero-mean Gaussian noise with $\sigma = 0.5$ pixels.

The estimated trajectory by LBA and BA, compared to ground truth and to pure IMU integration, is shown in Figure 1.13a, with position estimation errors given in Figure 1.13b. One can observe the fast drift of IMU-based dead reckoning, while both LBA and BA yield estimates close to ground truth with similar levels of accuracy. Note that only IMU and monocular cameras are used, *without* GPS or any additional sensors, producing position estimates with a typical estimation error of $5 - 10$ meters, with a highest estimation error of 20 meters.

While a similar estimation accuracy was obtained both by LBA and BA, processing time is different. The latter depends on the number of feature observations per frame γ, which affects the number of observed landmarks. We therefore discuss processing time for two different values of feature observations per frame, $\gamma = \{200, 500\}$, while performing exactly the same trajectory. In the former case, number of landmarks is 9.5k with total number of image observations of about 66k,

while in the latter case, number of landmarks and total number of image observations are 23k and 165k, respectively.

Processing time for these two cases is shown in Figures 1.14a-1.14b and summarized in Table 1.2. As seen, while BA exhibits lower processing time now and then, in particular when far from loop closure frames, the overall processing time is much smaller in the LBA case. One can clearly observe the spikes in BA, that are the result of massive variable re-elimination and re-linearization triggered by loop closures and proceeds for many frames afterwards. Overall, the average processing time per frame in the shown scenario for $\gamma = 200$ features is 0.27 and 0.59 seconds for LBA and BA, respectively. Increasing number feature observations per frame to $\gamma = 500$, leads to further difference in average processing time, as shown in Figure 1.14b: 0.57 and 1.95 seconds for LBA and BA. Thus, LBA is about 2 times faster, on average, than BA for $\gamma = 200$ features, and almost 5 times faster for $\gamma = 500$ features.

#Features per frame	#Landmarks	#Observations	Ave. Time [sec]		
			BA	**LBA**	Ratio
200	9.5k	66k	0.59	**0.27**	2.19
500	23k	165k	1.95	**0.57**	3.42

Table 1.2: Average processing time per camera frame.

1.6 Conclusions and Future Work

This paper focused on incremental light bundle adjustment (iLBA) [13], a structure-less bundle adjustment approach that reduces computational complexity by algebraically eliminating the 3D points using multiple view geometry constraints and utilizing an efficient incremental optimization - incremental smoothing. Our first contribution is a theoretical probabilistic analysis of iLBA, where we identified the root effects that may cause the underlying probability distribution of iLBA to be somewhat different from the probability distribution over camera poses that is calculated from full bundle adjustment. Using two real-imagery datasets we demonstrated that, in practice, these two probability distributions are very close in terms of the maximum a posteriori estimate and the estimated uncertainty.

The second contribution of this paper is an extension of iLBA to robotic navigation, where besides a camera sensor, additional sensors operating at different rates typically exist. In particular, we considered the problem of fusing information between high-rate inertial navigation sensors (IMU) and vision observations. Following the iLBA concept, our formulation avoids explicit estimation of camera-observed 3D points, and utilizes a recently developed technique for IMU pre-integration to significantly reduce the number of variables in the optimization. We
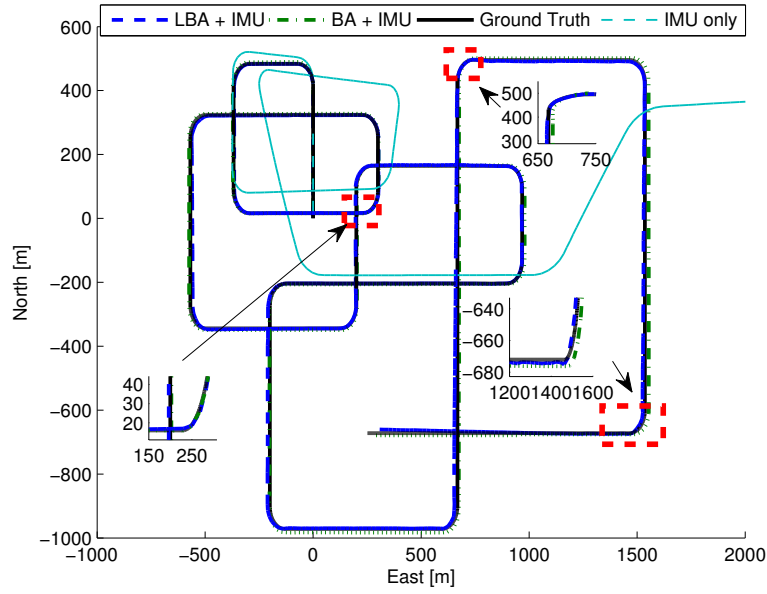
(a)
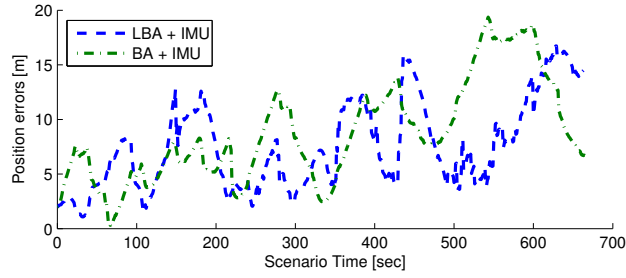
(b)

Fig. 1.13: (a) Top view of estimated trajectory. Inertial navigation quickly drifts while both LBA and BA result in bounded navigation errors over time; (b) Position estimation errors (norm).

demonstrated, based on a realistic synthetic aerial scenario, that iLBA for robotic navigation produces comparable state estimation accuracy to bundle adjustment formulation, where 3D points are explicitly inferred, while reducing average computational time by a factor of 2-3.5.

Future research will focus on developing approaches for optimally choosing past camera frames when adding new multi-view geometry constraints and on extensive experimental evaluation of the described application of iLBA to robotic navigation.
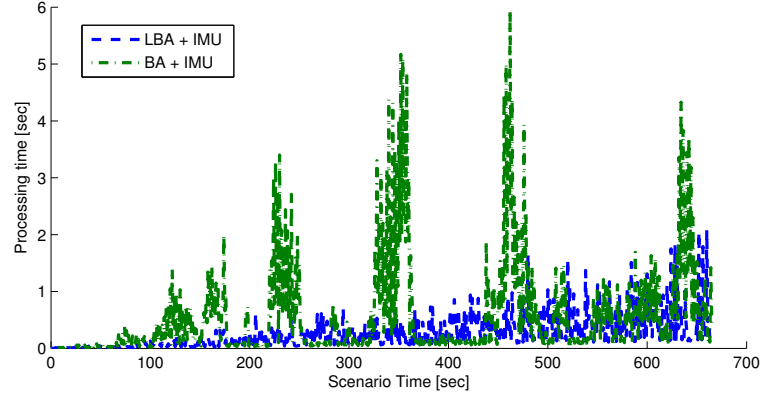
(a)
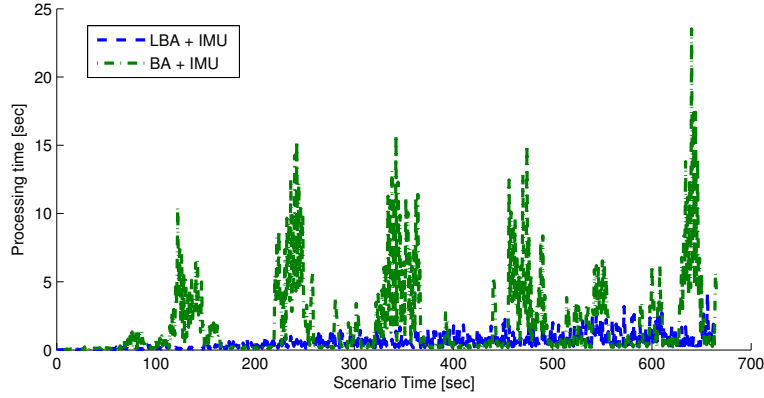
(b)

Fig. 1.14: (a) and (b) Processing timing comparison between the proposed method and bundle adjustment for 200 and 500 features per frame. Both methods use incremental smoothing and equivalent IMU factors.

Appendix

This appendix presents further details regarding the metric used to compare estimated camera pose uncertainty in Section 1.4.2.

To compare two covariance matrices Σ_1 and Σ_2, we define a discrepancy measure of the square roots of the traces of each covariance matrix,

$$discrepancy\left(\Sigma_1, \Sigma_2\right) \stackrel{\Delta}{=} c\left(\sqrt{\mathrm{tr}\left(\Sigma_1\right)} - \sqrt{\mathrm{tr}\left(\Sigma_2\right)}\right), \qquad (1.19)$$

where c is a scale factor that converts the unit-less 3D reconstructions into meters, which we determined by physically measuring the dataset collection area, or superimposing the trajectory onto a satellite image. We compute this separately for the blocks of the covariance matrices corresponding to rotation and translation. The units of the discrepancy are radians for rotation ($c = 1$) and meters for translation, with c properly determined to correct the reconstruction scale.

For example, to compare the Gaussian-approximated conditional density of LBA $p_{LBA}(x_i|x_j, Z)$ with covariance $\Sigma_{LBA}^{i|j}$ with that of BA $p(x_i|x_j, Z)$ with covariance $\Sigma_{BA}^{i|j}$, we compute $discrepancy\left(\Sigma_{LBA}^{i|j}, \Sigma_{BA}^{i|j}\right)$. Similarly for marginals $p_{LBA}(x_i|Z)$ and $p_{BA}(x_i|Z)$, we compute $discrepancy\left(\Sigma_{LBA}^{i}, \Sigma_{BA}^{i}\right)$. A positive discrepancy value means that the uncertainty estimate of LBA is conservative, whereas a negative discrepancy value means that the uncertainty estimate of LBA is overconfident.

References

1. S. Avidan and A. Shashua. Threading fundamental matrices. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(1):73–77, 2001.
2. D. Crandall, A. Owens, , N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3001–3008, 2011.
3. F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181–1203, Dec 2006.
4. R.M. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Trans. Robotics*, 22(6):1100–1114, Dec 2006.
5. J.A. Farrell. *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill, 2008.
6. Strasdat H., Montiel J. M. M., and Davison A. J. Scale drift-aware large scale monocular SLAM. In *Robotics: Science and Systems (RSS)*, Zaragoza, Spain, June 2010.
7. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
8. V. Ila, J. M. Porta, and J. Andrade-Cetto. Information-based compact Pose SLAM. *IEEE Trans. Robotics*, 26(1), 2010. In press.
9. V. Indelman. *Navigation Performance Enhancement Using Online Mosaicking*. PhD thesis, Technion - Israel Institute of Technology, 2011.
10. V. Indelman. Bundle adjustment without iterative structure estimation and its application to navigation. In *IEEE/ION Position Location and Navigation System (PLANS) Conference*, April 2012.
11. V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein. Real-time vision-aided localization and navigation based on three-view geometry. *IEEE Trans. Aerosp. Electron. Syst.*, 48(3):2239–2259, July 2012.
12. V. Indelman, A. Melim, and F. Dellaert. Incremental light bundle adjustment for robotics navigation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, November 2013.
13. V. Indelman, R. Roberts, C. Beall, and F. Dellaert. Incremental light bundle adjustment. In *British Machine Vision Conf. (BMVC)*, September 2012.
14. V. Indelman, R. Roberts, and F. Dellaert. Probabilistic analysis of incremental light bundle adjustment. In *IEEE Workshop on Robot Vision (WoRV)*, January 2013.
15. V. Indelman, S. Wiliams, M. Kaess, and F. Dellaert. Factor graph based incremental smoothing in inertial navigation systems. In *Intl. Conf. on Information Fusion, FUSION*, 2012.

16. V. Indelman, S. Wiliams, M. Kaess, and F. Dellaert. Information fusion in navigation systems via factor graph based incremental smoothing. *Robotics and Autonomous Systems*, 61(8):721–738, August 2013.
17. M. Kaess, V. Ila, R. Roberts, and F. Dellaert. The Bayes tree: An algorithmic foundation for probabilistic robot mapping. In *Intl. Workshop on the Algorithmic Foundations of Robotics*, Dec 2010.
18. M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31:217–236, Feb 2012.
19. M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, Dec 2008.
20. M. Kaess, S. Wiliams, V. Indelman, R. Roberts, J.J. Leonard, and F. Dellaert. Concurrent filtering and smoothing. In *Intl. Conf. on Information Fusion, FUSION*, 2012.
21. K. Konolige. Sparse sparse bundle adjustment. In *British Machine Vision Conf. (BMVC)*, September 2010.
22. K. Konolige and M. Agrawal. FrameSLAM: from bundle adjustment to realtime visual mapping. *IEEE Trans. Robotics*, 24(5):1066–1077, 2008.
23. F.R. Kschischang, B.J. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2), February 2001.
24. M.I. A. Lourakis and A.A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009.
25. F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, pages 333–349, Apr 1997.
26. T. Lupton and S. Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Trans. Robotics*, 28(1):61–76, Feb 2012.
27. Y. Ma, S. Soatto, J. Kosecka, and S.S. Sastry. *An Invitation to 3-D Vision*. Springer, 2004.
28. A.I. Mourikis and S.I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3565–3572, April 2007.
29. A.I. Mourikis and S.I. Roumeliotis. A dual-layer estimator architecture for long-term localization. In *Proc. of the Workshop on Visual Localization for Mobile Platforms at CVPR*, Anchorage, Alaska, June 2008.
30. K. Ni, D. Steedly, and F. Dellaert. Out-of-core bundle adjustment for large-scale 3D reconstruction. In *Intl. Conf. on Computer Vision (ICCV)*, Rio de Janeiro, October 2007.
31. A. L. Rodríguez, P. E. López de Teruel, and A. Ruiz. Reduced epipolar cost for accelerated incremental sfm. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3097–3104, June 2011.
32. N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.
33. N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *SIGGRAPH*, pages 835–846, 2006.
34. R. Steffen, J.-M. Frahm, and W. Förstner. Relative bundle adjustment based on trifocal constraints. In *ECCV Workshop on Reconstruction and Modeling of Large-Scale 3D Virtual Environments*, 2010.
35. R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-View Multibody Structure from Motion. *Intl. J. of Computer Vision*, 68(1):7–25, 2006.