Incremental Light Bundle Adjustment

Vadim Indelman
indelman@cc.gatech.edu
Richard Roberts
richard.roberts@gatech.edu
Chris Beall
cbeall3@gatech.edu
Frank Dellaert
dellaert@cc.gatech.edu

College of Computing,
Georgia Institute of Technology,
Atlanta, GA 30332, USA

Abstract

Fast and reliable bundle adjustment is essential in many applications such as mobile vision, augmented reality, and robotics. Two recent ideas to reduce the associated computational cost are structure-less SFM (structure from motion) and incremental smoothing. The former formulates the cost function in terms of multi-view constraints instead of re-projection errors, thereby eliminating the 3D structure from the optimization. The latter was developed in the SLAM (simultaneous localization and mapping) community and allows one to perform efficient incremental optimization, adaptively identifying the variables that need to be recomputed at each step.

In this paper we combine these two key ideas into a computationally efficient bundle adjustment method, and additionally introduce the use of three-view constraints to remedy commonly encountered degenerate camera motions. We formulate the problem in terms of a factor graph, and incrementally update a directed junction tree which keeps track of the current best solution. Typically, only a small fraction of the camera poses are recalculated in each optimization step, leading to a significant computational gain. If desired, all or some of the observed 3D points can be reconstructed based on the optimized camera poses. To deal with degenerate motions, we use both two and three-view constraints between camera poses, which allows us to maintain a consistent scale during straight-line trajectories. We validate our approach using synthetic and real-imagery datasets and compare it to standard bundle adjustment, in terms of performance, robustness and computational cost.

1 Introduction

In recent years several methods have been proposed for reducing the computational cost of bundle adjustment (BA) when processing a large number of images. Among these, methods that optimize the re-projection error cost function include [10], which proposes utilizing the sparse secondary structure of the Hessian, [19] which constructs a skeletal graph of a small subset of images and then incorporates the remaining images using pose estimation, and [14], where the authors propose to decouple the BA problem into several submaps that can

http://dx.doi.org/10.5244/C.26.134

be efficiently optimized in parallel. A thorough review of different aspects in BA can be found in [21].

This paper builds upon the recently introduced "structure-less" BA [6, 16, 17, 20], in which the camera poses are optimized without including structure in the iterative optimization procedure. In structure-less BA, the optimized cost function is based on multi-view constraints, instead of the conventional approach of minimizing re-projection errors. If required, the entire observed scene structure, or any part of it, can be reconstructed based on the optimized camera poses.

Early approaches suggested applying multi-view constraints for solving the SfM problem in a given sequence of images [3, 15]. The concept of avoiding structure estimation as an intermediate step has been proposed in several works. [1] used trifocal tensors to concatenate sequential fundamental matrices in a consistent manner, while [23] proposed local BA applied on a sliding window of triplets of images and correcting the image observations instead of estimating the 3D points that they represent. In [22], SfM is solved using a constrained least squares optimization, with the overall re-projection errors minimized subject to all independent two-view constraints.

While neither of the above methods have proposed a global optimization that is solely based on multi-view constraints, they paved the way to structure-less bundle adjustment [6, 16, 17, 20]. In [6, 20], the magnitude of corrections to the image observations is minimized, subject to satisfying the trifocal tensor and three-view constraints respectively. Rodríguez et al. [16, 17] obtained a significant improvement in the computational complexity by avoiding correcting the observations altogether.

This paper introduces an efficient and incremental structure-less bundle adjustment that is applicable both to SfM and robotics in large-scale environments. The first key component of the proposed method is a factor graph formulation for SfM problems, that allows applying a recently-developed approach for incremental smoothing [8, 9]. Using that approach, incremental optimization adaptively identifies which camera poses should be optimized. Therefore, in contrast to previously proposed incremental SfM and BA methods [16, 23], only a small portion of the camera poses is typically recalculated in each optimization step.

In addition, in this paper special attention is paid to degenerate camera configurations such as co-linear camera centers. To that end, we use the recently-developed formulation of three-view constraints [5, 7], which represents all the independent equations stemming from observing some unknown 3D point by three distinct views. Alternatively, one can apply trifocal constraints [4, 13] within the same framework. In contrast to using only epipolar geometry constraints for structure-less bundle adjustment [16, 17], the three-view and trifocal constraints allow consistent motion estimation even in a straight-line camera motion. Also, as opposed to conventional BA, previous approaches for structure-less BA are prone to fail due to degeneracies. In our approach, degeneracies are avoided by checking the condition of the essential matrix which is used to obtain camera rotation and translation initialization.

2 Structure-Less BA and Incremental Smoothing

2.1 Structure-Less Bundle Adjustment

We adopt the structure-less bundle adjustment formulation in [6, 20], where the cost function is written in terms of corrections made to the image observations, subject to satisfying applicable multi-view constraints. We consider a sequence of M views observing N 3D points, and denote the i^{th} camera pose by x_i and the measured and "fitted" image observations of the j^{th} observed 3D point by p_i^j and $\hat{\mathrm{p}}_i^j$, respectively. The cost function is then

$$J_{SLB}(\hat{\mathrm{x}},\hat{\mathrm{p}}) \doteq \sum_{i=1}^{N}\sum_{j=1}^{M}\left\|\mathrm{p}_i^j-\hat{\mathrm{p}}_i^j\right\|_{\Sigma}^2-2\lambda^T\mathrm{h}(\hat{\mathrm{x}},\hat{\mathrm{p}}) \tag{1}$$

with $\hat{\mathrm{x}}$ the estimated poses for all cameras, $\hat{\mathrm{p}}$ all observations across all views, Σ the measurement covariance, and where $\|\mathrm{a}\|_{\Sigma}^2 \doteq \mathrm{a}^T\Sigma^{-1}\mathrm{a}$ denotes the squared Mahalanobis distance. In Eq. (1), $\mathrm{h} \doteq \begin{bmatrix} \mathrm{h}_1^T & \ldots & \mathrm{h}_{N_h}^T \end{bmatrix}$ represents all multi-view constraints derived from the feature correspondences in the given sequence of views. The "*SLB*" subscript stands for structureless bundle adjustment. Each constraint h_i is a function of several camera poses and the image observations in the corresponding images. For simplicity, we assume a calibrated case, although the uncalibrated scenario can be handled as well.

2.2 Light Bundle Adjustment

To substantially reduce the computational complexity we follow the technique introduced in [16, 17], in that we avoid actually making corrections to the observations during optimization. Hence, instead of (1) we minimize the following algebraic cost function,

$$J_{LBA}(\hat{\mathrm{x}},\mathrm{p}) \doteq \sum_{i=1}^{N_h}\|\mathrm{h}_i(\hat{\mathrm{x}},\mathrm{p})\|_{\Sigma_i}^2 \tag{2}$$

where N_h is the number of constraints, and p are the *uncorrected* observations. The covariance matrices Σ_i in (2) are calculated as $\Sigma_i = A_i^T\Sigma A_i$, where A_i is the Jacobian of the constraint h_i with respect to corrections to the involved observations. The notation LBA stands for light bundle adjustment.

The objective functions above are typically minimized using a Levenberg-Marquardt non-linear optimization scheme, which necessitates linearizing the functions h_i above as well as re-computing the covariance matrices Σ_i. However, one can further reduce the computational complexity by calculating the covariances Σ_i only once and keeping them fixed throughout the entire optimization, which produces nearly-identical results.

As mentioned in the introduction, there are different possible formulations for the multi-view constraints. In this work, we are going to use the three-view constraints formulation, as is further discussed in Section 3.1.

2.3 Factor Graph Formulation and Incremental Smoothing

The second element of our approach is an incremental optimization scheme, borrowed from recent work in SLAM [8, 9]. This can be best explained within a graphical model framework, in particular using factor graphs [11], which we now review.

A factor graph is a bipartite graph $G = (\mathcal{F},\mathcal{X},\mathcal{E})$ with two types of nodes: *factor nodes* $f_\alpha \in \mathcal{F}$ and *variable nodes*, $x_i \in \mathcal{X}$. Edges $e_{\alpha i} \in \mathcal{E}$ between factor nodes and variable nodes are present if and only if the factor f_α involves the variable x_i. The factor graph G defines a factorization of the function $f(\mathcal{X})$ as

$$f(\mathcal{X}) = \prod_{\alpha} f_\alpha(\mathcal{X}_\alpha),$$

where $\mathcal{X}_\alpha \subset \mathcal{X}$ is the set of all variables x_j connected by an edge to factor f_α.

The optimization process corresponds to adjusting all the variables $\mathcal{X}$ to obtain a maximum a posteriori estimate

$$\hat{\mathcal{X}} = \arg\max_{\mathcal{X}} f(\mathcal{X}) = \arg\min_{\mathcal{X}} \left(-\log f(\mathcal{X})\right).$$

Assuming a Gaussian distribution, the above formulation is equivalent to a non-linear least-squares optimization, wherein a suitable cost function is minimized.

As discussed in Section 3.2, in structure-less BA the variables x_i are the camera poses ($\mathcal{X} \equiv \mathrm{x}$) while the factors represent the multi-view constraints. Each new image contributes factors that represent the added multi-view constraints between that image and the previously-processed images. The factor graph therefore encodes multi-view constraints that have been added for all these images.

Representing the system to be optimized in terms of factor graphs does not alter the nature of the underlying objective function. Indeed, linearizing the objective function can be done by linearizing each factor (non-linear least-squares term) separately, yielding a factor graph with linear Gaussian factors. Solving the resulting linear system can be seen as inference in a Gaussian factor graph. It can further be shown that the traditional QR or Cholesky solvers correspond to variable elimination in the resulting factor graph [2].

However, in an incremental setting the graphical framework yields distinct advantages. To be specific, performing a full batch optimization as each new view is processed is needlessly expensive. Typically, short-track feature matches encode valuable information for camera poses of only the recent past images. On the other hand, observing feature points seen previously by several views or re-observing a scene will typically involve optimizing many more camera poses.

In order to perform these updates efficiently without duplicating computation, we use the incremental smoothing algorithm introduced by Kaess et al. [9], which maintains a directed junction tree, called a *Bayes tree*, encoding the posterior density for the entire structure-less BA problem constructed so far. The key idea is to efficiently update only the subset of nodes of the Bayes tree affected by newly-added factors or by relinearization of variables with large linear updates. In order to enable efficient tree-based algorithms to perform these computations, and to reduce computational overhead, this method groups fully-connected variables into cliques that make up the nodes of the Bayes tree.

3 Incremental Light Bundle Adjustment (iLBA)

In this section we combine the two key methods described thus far, namely structure-less BA based on three-view constraints and incremental inference, into a single framework: incremental light bundle adjustment (*i*LBA).

3.1 Three-View Constraints

We use three-view constraints that were already proposed for structure-less bundle adjustment in [6], and allow consistent motion estimation even in straight trajectories, as opposed to only using two-view constraints [16]. The three-view constraints between some three views k, l, and m that observe the jth 3D point are given by [5]:

$$
\begin{aligned}
g_1 &= \mathrm{q}_k^j \cdot (\mathrm{t}_{k\to l} \times \mathrm{q}_l^j) && \text{(3a)}\\
g_2 &= \mathrm{q}_l^j \cdot (\mathrm{t}_{l\to m} \times \mathrm{q}_m^j) && \text{(3b)}\\
g_3 &= (\mathrm{q}_l^j \times \mathrm{q}_k^j) \cdot (\mathrm{q}_m^j \times \mathrm{t}_{l\to m}) - (\mathrm{q}_k^j \times \mathrm{t}_{k\to l}) \cdot (\mathrm{q}_m^j \times \mathrm{q}_l^j) && \text{(3c)}
\end{aligned}
$$

where $\mathrm{q}_i \doteq R_i^T K_i^{-1} \mathrm{p}$ for any view i and observation p, K_i is the calibration matrix of this view, R_i represents the rotation matrix from some arbitrary global frame to the i^{th} view's frame, and $\mathrm{t}_{i\to j}$ denotes the translation vector from view i to view j, expressed in the global frame. As seen, Eqs. (3a) and (3b) are the well-known epipolar geometry constraints, while Eq. (3c) facilitates maintaining a consistent scale for the translation vectors $\mathrm{t}_{l\to m}$ and $\mathrm{t}_{k\to l}$. These constraints were shown to be necessary and sufficient conditions for observing the same 3D point by these three views [5, 7]. The appendix provides further details regarding the relation of the three-view constraints to the standard trifocal tensor.

3.2 iLBA using Three-View Constraints

While the explicit measurement model assumed in Section 2.3 is appropriate for the projection equations used in conventional BA, the three-view constraints (3) represent an implicit measurement model of the form $\mathrm{g}(\mathcal{X}_i, \mathrm{z}_i)$. In our case, the variables $\mathcal{X}_i$ are the appropriate camera poses and the measurements z_i are the matching observations in these images.

The equivalent factor for optimizing the cost function J_{LBA} (2) using an implicit measurement model and assuming a Gaussian distribution of the residual error (cf. Section 2.1) is defined as

$$f_i(\mathcal{X}_i) \doteq \exp\left(\|\mathrm{g}(\mathcal{X}_i, \mathrm{z}_i)\|_\Sigma^2\right)$$

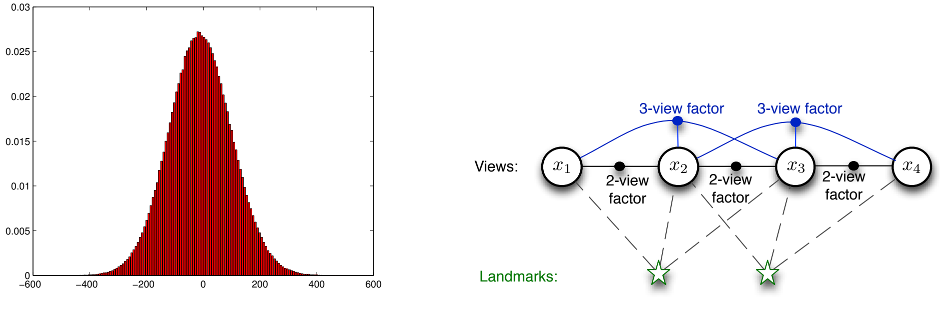
with a measurement covariance matrix Σ. Figure 1a presents the distribution of this error for a three-view constraint (3c), obtained by introducing zero-mean Gaussian perturbations on ideal synthetic observations and camera poses. It can be seen that the Gaussian distribution assumption is indeed valid. A similar distribution is also obtained for the two-view constraint.

In practice, each of the constraints in (3) is added as a separate factor for each (unknown) 3D point that is observed by some three views. For each additional view k that observes the same 3D point, we only add two of these constraints between that view and some two earlier views l and m: a two-view constraint between view k and either view l or view m, and a three-view constraint (3c) between these three views. The reason for not adding the second two-view constraint (between views l and m) is that this constraint was actually already used when processing these past views. We add a standard two-view constraint in case a 3D point is observed by only two views.

Next we explicitly write factor formulations for the two-view and three-view constraints (3) for optimizing the cost function (2). Figure 1b illustrates a factor graph using two- and three-view constraints in a basic example.

A two-view constraint between some two views k and l with an observation correspondence $\mathrm{p}_k, \mathrm{p}_l$ is given by $g_{2-view}(\mathrm{x}_k, \mathrm{x}_l, \mathrm{p}_k, \mathrm{p}_l) \doteq \mathrm{q}_k \cdot (\mathrm{t}_{k\to l} \times \mathrm{q}_l) \equiv g_1$. Since we only optimize over camera poses, the equivalent factor is

$$f_{2-view}(\mathrm{x}_k, \mathrm{x}_l) \doteq \exp\left(\|g_{2-view}(\mathrm{x}_k, \mathrm{x}_l, \mathrm{p}_k, \mathrm{p}_l)\|_{\Sigma_{2-view}}^2\right)$$



(a) (b)

Figure 1: (a) Distribution of the residual error in a three-view constraint. The Gaussian-like nature of the distribution legitimates the usage of the cost function J_{LBA} over J_{SLB}. (b) A factor graph representation for a simple example of 4 cameras observing 2 landmarks. Two-view and three-view factors are added instead of projection factors. Landmark observations are denoted by dashed lines.

where covariance $\Sigma_{2-view} \doteq \left(\nabla_{\mathrm{p}_k,\mathrm{p}_l} g_{2-view}\right)^T \Sigma \left(\nabla_{\mathrm{p}_k,\mathrm{p}_l} g_{2-view}\right)$ is used to calculate the Mahalonobis distance.

The three-view constraint (3c) for some three views k,l and m, can be similarly written as $g_{3-view}\left(\mathrm{x}_k,\mathrm{x}_l,\mathrm{x}_m,\mathrm{p}_k,\mathrm{p}_l,\mathrm{p}_m\right) \equiv g_3$ and therefore the equivalent factor representation is $f_{3-view}\left(\mathrm{x}_k,\mathrm{x}_l,\mathrm{x}_m\right) \doteq \exp\left(\|g_{3-view}\left(.\right)\|^2_{\Sigma_{3-view}}\right)$ with the appropriate covariance matrix Σ_{3-view}.

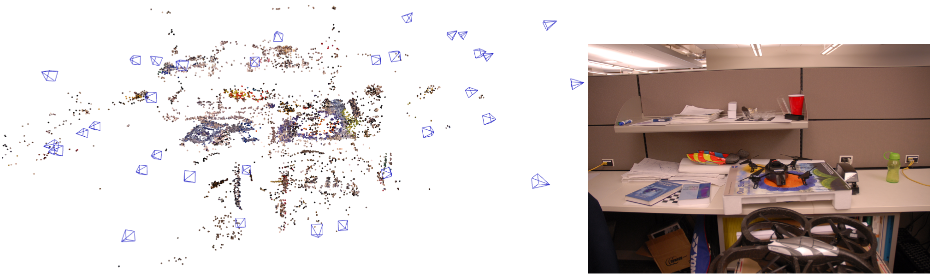
Processing a new incoming image involves constructing the above two-view and three-view factors from the appropriate constraints and performing an incremental inference (instead of a full optimization) as discussed in Section 2.3.

Calculating an initial pose for the new image given some previously-processed two overlapping images involves two steps: we first calculate an initial estimate the relative motion between the new image and one of these previous images by calculating the essential matrix and extracting the motion parameters from it [4] (an initial motion estimate can also be found by solving the two-view constraints). Since the translation is known only up to a scale, we apply Eq. (3c) to calculate a consistent magnitude of the translation vector while considering the rest of the motion parameters fixed.

4 Results

We present results demonstrating *i*LBA on two indoor datasets that were collected in our lab, *cubicle* and *straight*, and on a synthetic dataset, *circle*. In the *cubicle* dataset the camera observed a cubicle desk from various angles and distances, while the *straight* dataset consists of a straight forward motion of a forward-facing camera. In the synthetic *circle* dataset the cameras are positioned along a circle pointing inwards, while the landmarks are scattered in the center area. Table 1 provides additional information regarding the number of views (N) and landmarks (M), as well as the number of total observations in each dataset.

Figure 2 shows one of the images in the *cubicle* dataset and the optimized camera poses and reconstructed structure using *i*LBA. The structure was reconstructed based on the optimized camera poses after *i*LBA has converged.



(a) (b)

Figure 2: (a) Optimized camera poses and reconstructed structure in the *cubicle* dataset. (b) One of the images in a *cubicle* dataset.

The proposed method is compared, in terms of computational cost and accuracy, to the following methods:

- *i*LBAΣ: *i*LBA with the covariance Σ_i re-calculated each time a linearization occurs, as opposed to *i*LBA in which it is only calculated once at the beginning (cf. Section 2.1)
- BA: conventional bundle adjustment
- SLB using three-view constraints: similar to the cost function used in [20]

We also present a comparison between an incremental smoothing, as discussed in Section 2.3, and a naïve incremental batch optimization. To ensure a fair comparison, convergence criteria and maximum nonlinear iteration parameters were set to the same values in both methods.

All results were obtained on a 2.2 GHz Core i7 laptop using a single-threaded implementation. The different structure-less BA methods, including *i*LBA, were implemented using custom two-view and three-view factors and optimized using the GTSAM factor graph optimization library[1] [2, 9], while conventional BA used projection factors provided by GTSAM.

Image correspondences, as well as the calibration matrices, were obtained by first running bundler[2] [18] on each indoor dataset. In our method the incremental initializations of the camera poses were calculated by estimating essential matrices between pairs of images, as described in Section 3.2. Degenerate and ill-conditioned camera pairs were identified by having either an under-determined or invalid essential matrix, and our method avoids adding constraints between these degenerate pairs, instead choosing well-conditioned pairs. Initial values for landmarks, required for a conventional BA, were computed using triangulation. The bundler solution was not used to initialize any camera poses or landmarks. Initial values of camera poses for the synthetic dataset *circle* were obtained by corrupting the ground truth with a Gaussian zero-mean errors with a standard deviation of $\sigma = 10$ meters for the position and $\sigma = 0.5$ degrees for rotation terms in each axis.

Table 1 shows re-projection errors (mean μ and standard deviation σ) for the compared methods. All the results shown in this table were obtained by performing the described incremental smoothing scheme. For structure-less BA methods, the re-projection errors were calculated after structure reconstruction was performed based on the optimized camera poses.

[1] http://tinyurl.com/gtsam.

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

Dataset	BA	*i*LBA	*i*LBAΣ	SLB	N, M, #Obsrv
Cubicle	1.981 (μ) 1.6301 (σ)	2.1017 (μ) 1.8364 (σ)	2.0253 (μ) 1.742 (σ)	1.9193 (μ) 1.6294 (σ)	33, 11066, 36277
Straight	0.519 (μ) 0.4852 (σ)	0.5434 (μ) 0.5127 (σ)	0.5407 (μ) 0.5098 (σ)	0.5232 (μ) 0.4870 (σ)	14, 4227, 14019
Circle (synthetic)	0.6186 (μ) 0.3220 (σ)	0.6244 (μ) 0.3253 (σ)	0.6235 (μ) 0.3246 (σ)	0.6209 (μ) 0.3235 (σ)	120, 500, 58564

Table 1: Re-projection errors using incremental smoothing in all methods.

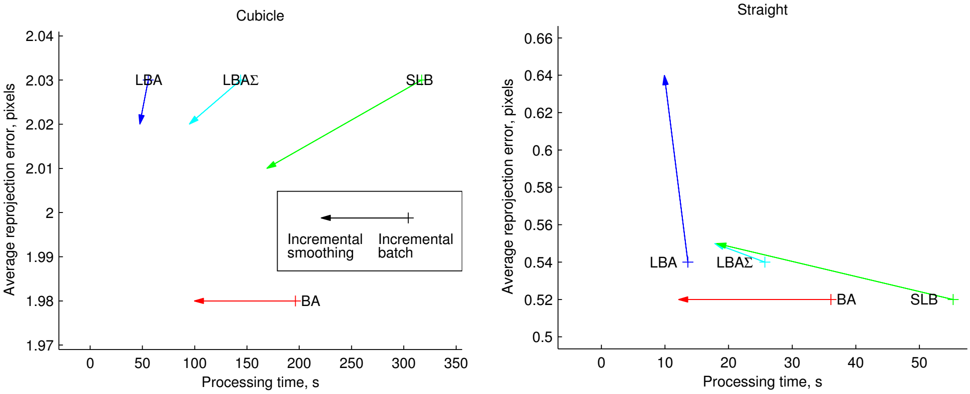


Figure 3: : Comparing incremental smoothing with incremental batch, in terms of average reprojection error versus processing time. Average reprojection error magnitudes are very similar, but incremental smoothing significantly reduces processing time.

Overall, similar values of re-projection errors were obtained for all methods. BA produced the best results, which are slightly degraded for SLB, *i*LBAΣ and *i*LBA.

Although not explicitly shown, structure-less BA with only two-view constraints [16] is degenerate in the *straight* dataset, where the camera centers are close to co-linear, and thus fails to run to completion. In such configurations, the magnitude of the translation vector cannot be recovered without using three-view or trifocal constraints and thus the problem is under-constrained. In contrast, *i*LBA continues to produce consistent results even in this challenging dataset.

Figure 3 shows the trade-off between the accuracy (re-projection errors) and computational cost (timing) for the considered different approaches for the *cubicle* dataset, in particular demonstrating the computational gain when using incremental smoothing instead of naïve incremental batch optimization. For structure-less BA methods the presented timing results include both the camera-pose optimization and the structure reconstruction.

As seen, *i*LBA yields significantly better timing results compared to other methods, with negligible reduction in accuracy. The incremental smoothing approach improves timing results of all methods, compared to incremental batch optimization, and has only a small effect on the accuracy. In particular, processing time for the *cubicle* dataset is 47.5 seconds for incremental LBA and structure reconstruction, as opposed to 99.5 seconds in incremental BA. The fact that the SLB formulation [20] is slower than all the other methods, including BA, is not surprising, since in that formulation the observations are being corrected and therefore

Dataset	BA	Structure-less BA			
		*i*LBA	*i*LBAΣ	SLB	structure recon.
Cubicle	99.5	29.1	73.7	147.3	18.4
Straight	12.1	3.0	10.0	10.5	6.9
Circle (synthetic)	>2hr	131.8	301.6	>2hr	3.8

Table 2: Computational cost (in seconds): all methods used incremental smoothing; the structure reconstruction phase is nearly-identical in all structure-less BA methods.

each landmark is represented by all its observations. Therefore, the optimization performed in SLB involves more variables than in BA.

Additional details for the computational cost are given in Table 2, in which all the methods use the incremental smoothing scheme and the results for structure-less BA optimization and the structure reconstruction are shown separately. The cost for structure reconstruction is very similar in all structure-reconstruction methods. One can observe that the computational cost in both BA and SLB formulations is much higher than in *i*LBA in the synthetic *circle* dataset. This can be explained by the dense nature of this dataset, where each 3D point is observed by all cameras.

Further computational gain can be obtained by calculating a reduced measurement matrix [4] from the applied multi-view constraints, as was already presented in [16] for two-view constraints and can be applied to three-view constraints as well.

5 Conclusions

This paper described a fast incremental structure-less bundle adjustment technique, advancing previous work on structure-less bundle adjustment in two main aspects. First, the cost function was formulated using the recently-developed three-view constraints which allow for consistent motion estimation even in co-linear camera configurations. Secondly, a factor graph representation of the optimization problem was introduced, allowing for the application of incremental smoothing. As opposed to previous incremental bundle adjustment methods, in the proposed method only part of the camera poses, that are adaptively identified, participate in the optimization process, leading to reduced computational complexity. Degenerate configurations are properly handled, as well. Future work includes the validation of the proposed approach on much larger-scale datasets, where computational time-savings should be even more significant.

Appendix

In this appendix we relate between the well-known trifocal constraints and the three-view constraints (3). Assume some 3D point X is observed by several views. The image projection of this landmark for each of these views is given by $\lambda_i \mathrm{p}_i = P_i \mathrm{X}$, where λ_i is the unknown scale parameter and $P_i = K_i \left[\begin{array}{cc} R_i & \mathrm{t}_i \end{array} \right]$ is the projection matrix of the i^{th} view.

Choosing the reference frame to be the first view, the projection equations turn into $\lambda_1 K_1^{-1} \mathrm{p}_1 = \check{\mathrm{X}}$ and $\lambda_i K_i^{-1} \mathrm{p}_i = R_i \check{\mathrm{X}} + \mathrm{t_i}$ for $i > 1$. Here $\check{\mathrm{X}}$ denotes inhomogeneous coordinates of the 3D point X. Substituting the former equation into the latter equation eliminates the 3D point, yielding

$$\lambda_i K_i^{-1} \mathrm{p}_i = R_i \lambda_1 K_1^{-1} \mathrm{p}_1 + \mathrm{t_i} \, , \, i > 1 \tag{4}$$

From this point the derivation of three-view and trifocal constraints differs. In the former case, a matrix is constructed from Eq. (4) for the first view and two other views i and j:

$$\begin{bmatrix} R_i K_1^{-1} \mathrm{p}_1 & -K_i^{-1} \mathrm{p}_i & 0 & \mathrm{t_i} \\ R_j K_1^{-1} \mathrm{p}_1 & 0 & -K_j^{-1} \mathrm{p}_j & \mathrm{t_j} \end{bmatrix} \begin{bmatrix} \lambda_1 & \lambda_i & \lambda_j & 1 \end{bmatrix}^T = 0 \tag{5}$$

while in the case of trifocal constraints, the additional scale parameter in Eq. (4) is eliminated by cross multiplying with $K_i^{-1} \mathrm{p}_i$. Representing the resulting equations in a matrix formulation yields the so-called multi-view matrix [12]:

$$\begin{bmatrix} \left[K_2^{-1} \mathrm{p}_2 \right]_\times R_2 K_1^{-1} \mathrm{p}_1 & \left[K_2^{-1} \mathrm{p}_2 \right]_\times \mathrm{t}_2 \\ \left[K_3^{-1} \mathrm{p}_3 \right]_\times R_3 K_1^{-1} \mathrm{p}_1 & \left[K_3^{-1} \mathrm{p}_3 \right]_\times \mathrm{t}_3 \\ \vdots & \vdots \end{bmatrix} \begin{pmatrix} \lambda_1 \\ 1 \end{pmatrix} = 0 \tag{6}$$

Enforcing the rank-deficiency condition on the two matrices in Eqs. (5) and (6) yields the three-view and the trifocal constraints [5, 12]. Although Eq. (6) contains expressions for all the views, the resulting constraints relate only between triplets of views.

References

[1] S. Avidan and A. Shashua. Threading fundamental matrices. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(1):73–77, 2001.

[2] 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.

[3] A. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Eur. Conf. on Computer Vision (ECCV)*, pages 311–326, 1998.

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

[5] V. Indelman. *Navigation Performance Enhancement Using Online Mosaicking*. PhD thesis, Technion - Israel Institute of Technology, 2011.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] K. Konolige. Sparse sparse bundle adjustment. In *BMVC*, pages 1–11, September 2010.

[11] 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.

[12] Y. Ma, K. Huang, R. Vidal, J. Košecká, and S. Sastry. Rank conditions on the multiple-view matrix. 59(2):115–137, September 2004.

[13] Y. Ma, S. Soatto, J. Kosecka, and S.S. Sastry. *An Invitation to 3-D Vision*. Springer, 2004.

[14] 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.

[15] D. Nistér. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *Eur. Conf. on Computer Vision (ECCV)*, pages 649–663, 2000.

[16] 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.

[17] A. L. Rodríguez, P. E. López de Teruel, and A. Ruiz. GEA optimization for live structureless motion estimation. In *First Intl. Workshop on Live Dense Reconstruction from Moving Cameras*, pages 715–718, 2011.

[18] N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *SIGGRAPH*, pages 835–846, 2006.

[19] 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.

[20] 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.

[21] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *LNCS*, pages 298–372. Springer Verlag, 2000.

[22] R. Vidal, Y. Ma, S. Hsu, and S. Sastry. Optimal motion estimation from multiview normalized epipolar constraint. In *Intl. Conf. on Computer Vision (ICCV)*, volume 1, pages 34–41, 2001.

[23] Z. Zhang and Y. Shan. Incremental motion estimation through local bundle adjustment. Technical Report MSR-TR-01-54, Microsoft Research, May 2001.