# Bundle Adjustment Without Iterative Structure Estimation and its Application to Navigation

Vadim Indelman*

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

***Abstract*—This paper describes a new approach for bundle adjustment (BA), which is based on a non-linear optimization of only the camera pose for all the views in a given sequence of images and does not involve iterative structure estimation. If required, structure reconstruction can be performed based on the camera matrices after convergence of the optimization process. Instead of applying the projection equations, the cost function being optimized in the suggested approach is based on the three-view geometry constraints that should be satisfied for any three views with a common overlapping area. Significant reduction in computational complexity is obtained compared to a standard BA, since the number of unknown parameters participating in the iterative optimization is much smaller. The optimization problem is formulated relative to the camera pose of the first view, as commonly used in robotics navigation aplications. The proposed method is demonstrated on a publiclly available dataset of real images and the optimized camera pose and the recovered structure are compared to the ground truth.**

## I. Introduction

Bundle adjustment (BA) has been at the focus of many research efforts in the past several decades. The goal in BA is to estimate the camera matrices and the 3D coordinates of observed landmarks given a set of images. Data association, namely, the pixels correspondence between the given views, is usually assumed to be solved.

The general approach for solving the BA problem is to perform a non-linear optimization, minimizing the error between the measured and the projected image coordinates of the 3D landmarks. The optimization process involves simultaneously solving for all the camera matrices (pose) and the 3D landmark coordinates (structure). Taking into account the sparse nature of the problem allows to significantly reduce the involved computational burden. A review of different theoretic and implementation aspects of BA can be found in [18].

Performing BA on long sequences of views is not an easy task even for now-days computers, both due to a high computational cost and to numerical stability issues. Several approaches were proposed for reducing the involved computational burden [20], [10], [13], [9] while approximating different aspects of the overall re-projection error cost function. For example, Zhang and Shan [20] applied bundle adjustment to a sliding window of triples of images; In [13], only the last cameras and the observed 3D points in these and some of the earlier cameras, are optimized; Konolige and Agrawal [9] proposed to maintain only part of the images, while the rest of the images and the observed 3D points in these images are accounted for by marginalization. In [15], a relative formulation was suggested, allowing adjusting only part of the variables in the BA optimization.

In this paper, a new BA formulation is proposed, in which the optimization involves only the camera pose along the given sequence of views. This approach, referred as light bundle adjustment (LBA), reduces the number of unknowns in the optimization and leads to a significant computational gain, since as opposed to conventional BA, the observed landmarks are no longer part of the optimization. The observed landmarks, or a partial set of them, can be estimated based on the optimized camera poses, using structure reconstruction techniques [4]. The obtained solution for camera matrices can also be used as initial conditions in the full BA optimization.

Instead of minimizing the overall re-projection error, the cost function minimized in LBA is formulated using multi-view constraints [4], [12]. Application of multi-view constraints for BA-related problems has been already proposed in the literature. For example, [3] suggested estimating the trifocal tensors between consecutive triplets of images as well as the observed 3D points in each such triplet, followed by registration of the triplets sequence (and the involved structure) into a single reference frame. Liu et al. [10] considered structure reconstruction from a given sequence of images with known camera pose, and proposed correcting the image coordinates of corresponding points by applying epipolar constraints between pairs of images, thereby

yielding an improvement in estimation of 3D points. Avidan and Shashua [1] suggested using trifocal tensors for consistently concatenating camera matrices, and applied their method on a sliding window of triplets of images.

As opposed to the above-cited works, methods for solving the full BA problem using multi-view constraints, and in particular capable of incorporating loop closures, are much less in common. The work by Steffen et al. [16], is arguably the most relevant to this paper. Similarly to [16], the cost function is formulated without any structure parameters, and the optimization is applied on the whole sequence of images, thereby naturally accommodating any loop closures. However, while Steffen et al. [16] use trifocal constraints, the current paper applies the recently-developed three-view constraints [5] in the optimization formulation. These constraints are attractive due to their simple form and were already proposed for real time vision-aided navigation [6] and cooperative navigation [8].

The remainder of this paper is organized as follows. Section III introduces relevant notations and formulates the problem; The next section reviews the optimization performed in a conventional BA; The proposed approach for light bundle adjustment is elaborated in Section IV-A, which formulates the new cost function and applies the mentioned three-view constraints; Section V discusses structure reconstruction. Results, demonstrating the method on a dataset of real images, are given in Section VI, while Section VII suggests concluding remarks.

## II. Problem Formulation

Consider a given set of $N$ partially overlapping views and $M$ unknown landmarks that are observed in some of these images. For a pinhole camera model, the image and the world coordinates of the $i$th landmark are related by the projection equation [4]:

$$\mathbf{p}_{ij} = K_j \left[ \begin{array}{cc} R_{r\to j} & \mathbf{t}^j_{j\to r} \end{array} \right] \mathbf{P}^r_i = M_j \mathbf{P}^r_i \qquad (1)$$

where $K_j$ is the camera calibration matrix of the $j$th camera, $\mathbf{P}^r_i = [X^r_i,\ Y^r_i,\ Z^r_i,\ 1]^T$ are the homogeneous 3D coordinates of the $i$th landmark, expressed in some reference frame $r$, and $\mathbf{p}_{ij} = [u_{ij},\ v_{ij},\ 1]^T$ are the corresponding homogeneous image coordinates. The matrix $R_{r\to j}$ is the rotation from the reference frame $r$ to the camera frame of the $j$th view, while $\mathbf{t}^j_{j\to r}$ is the translation vector between these two frames, expressed in the coordinate system of the $j$th view. Define the camera pose of the $j$th view as

$$\mathbf{x}_j \triangleq \left[ \begin{array}{c} \mathbf{t}^j_{j\to r} \\ \mathbf{\Psi}_{r\to j} \end{array} \right]$$

where $\mathbf{\Psi}_{r\to j}$ are the angles representing the rotation matrix $R_{r\to j}$.

In this paper it is assumed that the camera calibration matrices are known and the correspondence problem is solved. Therefore, each 3D point $\mathbf{P}_i$ is associated with a set of image pixels $\{\mathbf{p}_{ij}\}$ from appropriate views $j$.

Another assumption in the current work, is that some initial values for the camera pose are given for each of the $N$ views. These initial values can be obtained either by estimating the relative motion between consecutive views (e. g., by applying the 5-point algorithm [14]), be encoded as a meta-data to each of the images (e.g., images obtained from the internet), or obtained from a navigation system.

Our general objective is to optimize the camera pose $\mathbf{x}_j$ along the sequence of views ($j \in \{1, \dots N\}$) and the observed landmarks $\mathbf{P}_i$ (with $i \in \{1, \dots M\}$) given the image observations $\mathbf{p}_{ij}$ in the appropriate images.

## III. Bundle Adjustment

Conventional bundle adjustment [18] optimizes the overall re-projection error, which is defined for $N$ views observing $M$ landmarks as

$$J^{BA} \triangleq \sum_{j=1}^{N} \sum_{i=1}^{M} d(\mathbf{p}_{ij}, \mathbf{p}^{proj}_{ij}), \qquad (2)$$

where $\mathbf{p}_{ij}$ and $\mathbf{p}^{proj}_{ij}$ are the measured and predicted image coordinates. The prediction is made by the projection operator $\mathbf{Proj}$, defined by Eq. (1). In the above equation, the operator $d(.)$ represents some cost function. Assuming a Gaussian noise distribution, the cost function is defined as the squared Mahalanobis distance $d(e) \triangleq e^T \Sigma^{-1} e$, with $\Sigma$ being the estimated measurement covariance. Eq. (2) can therefore be written explicitly as

$$J^{BA} = \sum_{j=1}^{N} \sum_{i=1}^{M} \left\| \mathbf{p}_{ij} - \mathbf{Proj}\left(\mathbf{x}_j, \mathbf{P}_i\right) \right\|^2_{\Sigma} \qquad (3)$$

Eqs. (2)-(3) implicitly assume a zero re-projection error of any 3D point, that is not observed in some view.

The above cost function $J^{BA}$ is being optimized for all the unknown parameters, that can be represented in the following state vector:

$$\mathbf{x}^{BA} = \left[ \begin{array}{cccccc} \mathbf{x}_1^T & \dots & \mathbf{x}_N^T & \mathbf{P}_1^T & \dots & \mathbf{P}_M^T \end{array} \right]^T,$$

and the total number of the unknown variables is $6N + 3M$, i.e. $\mathbf{x}^{BA} \in \mathbb{R}^{(6N+3M)\times 1}$.

## IV. Light Bundle Adjustment

Similarly to [19], [10], [11], it is proposed to approximate the re-projection error $\boldsymbol{\epsilon}_{ij} \triangleq \mathbf{p}_{ij} - \mathbf{Proj}(\mathbf{x}_j, \mathbf{P}_i)$ by $\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij}$ where $\hat{\mathbf{p}}_{ij}$ are the predicted image coordinates satisfying a non-linear function $\mathbf{h}$. This function represents multi-view constraints, and therefore it does not contain any structure parameters. Using a Lagrange multiplier vector $\boldsymbol{\lambda}$, the new cost function can be written as

$$J^{LBA} \triangleq \sum_{j=1}^{N}\sum_{i=1}^{M} \left\|\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij}\right\|^2_{\Sigma_{ij}} - 2\boldsymbol{\lambda}^T \mathbf{h}(\hat{\mathbf{x}}, \hat{\mathbf{p}}) \quad (4)$$

where $\hat{\mathbf{x}}$ is the estimated camera pose of all the involved views, defined as

$$\mathbf{x}^{LBA} = \begin{bmatrix} \mathbf{x}_1^T & \dots & \mathbf{x}_N^T \end{bmatrix}^T \in \mathbb{R}^{6N\times 1}, \quad (5)$$

and $\hat{\mathbf{p}}$ are the corrected image coordinates in all the views so that the constraints-function $\mathbf{h}$ is satisfied:

$$\mathbf{h}(\hat{\mathbf{x}}, \hat{\mathbf{p}}) = \mathbf{0}. \quad (6)$$

The cost function $J^{LBA}$ can be compactly written as

$$J^{LBA} = \|\mathbf{p} - \hat{\mathbf{p}}\|^2_{\Sigma} - 2\boldsymbol{\lambda}^T \mathbf{h}(\hat{\mathbf{x}}, \hat{\mathbf{p}}). \quad (7)$$

The cost function $J^{LBA}$ (7) involves optimizing *only* for the pose parameters, since it does not contain any structure parameters. Therefore, the overall number of unknowns is *reduced to* $6N$ represented by the state vector $\mathbf{x}^{LBA}$, as opposed to the $6N + 3M$ unknowns in conventional BA (cf. Section III).

The non-linear optimization involved with minimizing the cost function $J^{LBA}$ is described in the Appendix.

While different formulations can be used for multi-view constraints [4], [12], [19], represented by the function $\mathbf{h}$ in Eqs. (4)-(7), a recently-developed formulation for three-view constraints [5] is suggested in the next section.

### A. LBA Using Three-View Constraints

The three-view constrains can be written, for any three views $k, l, m \in \{1, \dots, N\}$ with a common overlapping area, as [5]:

$$\bar{\mathbf{q}}_k^T (\bar{\mathbf{t}}_{k\to l} \times \bar{\mathbf{q}}_l) = 0 \quad (8)$$

$$\bar{\mathbf{q}}_l^T (\bar{\mathbf{t}}_{l\to m} \times \bar{\mathbf{q}}_m) = 0 \quad (9)$$

$$(\bar{\mathbf{q}}_l \times \bar{\mathbf{q}}_k) \cdot (\bar{\mathbf{q}}_m \times \bar{\mathbf{t}}_{l\to m}) = (\bar{\mathbf{q}}_k \times \bar{\mathbf{t}}_{k\to l}) \cdot (\bar{\mathbf{q}}_m \times \bar{\mathbf{q}}_l) \quad (10)$$

where $\mathbf{q}_k, \mathbf{q}_l$ and $\mathbf{q}_m$ are the line-of-sight vectors of the corresponding pixels in the three views, and $\mathbf{t}_{k\to l}$ and $\mathbf{t}_{l\to m}$ are the translation vectors between these views. The notation $\bar{\mathbf{a}}$ denotes the ideal value of some vector $\mathbf{a}$. The line-of-sight vector $\mathbf{q}$ for a given pixel $\mathbf{p}$ can be calculated in the camera system as

$$\mathbf{q} = K^{-1}\mathbf{p}.$$

Appropriate rotation matrices should be used for expressing all the vectors in Eqs. (8)-(10) in the same coordinate system, which can be chosen arbitrary.

Consequently, Eqs. (8)-(10) can be expressed as Eq. (6), which is part of the LBA cost function $J^{LBA}$ (cf. Eq. (4)).

The three-view constraints consist of the well-known epipolar geometry constraints (8)-(9) [4], and of an additional constraint (10) that allows to maintain a consistent scale between the three given views. As proven in [5], [7], these constraints are necessary and sufficient conditions for a general scene observed by the given three views.

Similar to a conventional BA, the overall set of equations in LBA is rank-deficient (or in other words, unobservable) and has 7 degrees of freedom. Therefore, estimating the absolute camera matrices is not trivial and requires a proper regularization. Instead of trying optimizing the camera poses with respect to some reference frame, it is suggested to estimate the camera motion of all views relative to one of the views in the sequence, thereby fixing 6 of the 7 degrees of freedom. Further discussion regarding the additional degree-of-freedom is provided in Section IV-E.

### B. Relative Formulation

As common in robotics applications, the camera poses are expressed, in this section, relative to the first camera pose. The relative state vector for the $N$ views in the sequence is defined as:

$$\mathbf{x}^{rel} \doteq \begin{bmatrix} \mathbf{x}_{1\to 2}^T & \cdots & \mathbf{x}_{1\to N}^T \end{bmatrix}^T \in \mathbb{R}^{6(N-1)\times 1} \quad (11)$$

with

$$\mathbf{x}_{1\to j} \triangleq \begin{bmatrix} \mathbf{t}_{1\to j}^1 \\ \boldsymbol{\Psi}_{1\to j} \end{bmatrix}, \quad j \in [2, N] \quad (12)$$

Different variations of relative formulation have been also proposed in several BA works, including [15], [16]. In particular, the equivalent projection equations to the above relative formulation, are [4]

$$M_j = K_j \begin{bmatrix} R_{1\to j} & \mathbf{t}_{j\to 1}^j \end{bmatrix} \quad (13)$$

with $j \in \{2,\ldots,N\}$ and $M_1 = K_1 \begin{bmatrix} I & \mathbf{0} \end{bmatrix}$ for the first view.

It is now possible to rewrite the three-view constraints (8)-(10) in terms of the state vector $\mathbf{x}^{rel}$. Note that the relative translation terms in Eq. (12) are expressed in the coordinate system of the first view. Also, the reference system $r$ does not participate in this relative formulation.

Due to image noise and imperfect estimation of $\mathbf{x}^{rel}$, the constraints (8)-(10) will not be satisfied: there will always be some residual error. The three-view constraints (8)-(10), for some three views $k,l,m \in \{1,\ldots,N\}$ observing the $i$th landmark ($i \in \{1,\ldots,M\}$), can be expressed as

$$z_1 \triangleq \mathbf{q}_k^T(\mathbf{t}_{k\to l} \times \mathbf{q}_l) \tag{14}$$

$$z_2 \triangleq \mathbf{q}_l^T(\mathbf{t}_{l\to m} \times \mathbf{q}_m) \tag{15}$$

$$z_3 \triangleq (\mathbf{q}_l \times \mathbf{q}_k)^T(\mathbf{q}_m \times \mathbf{t}_{l\to m}) - (\mathbf{q}_k \times \mathbf{t}_{k\to l})^T(\mathbf{q}_m \times \mathbf{q}_l) \tag{16}$$

with

$$\mathbf{t}_{k\to l} = \mathbf{t}^1_{1\to l} - \mathbf{t}^1_{1\to k} \tag{17}$$

$$\mathbf{t}_{l\to m} = \mathbf{t}^1_{1\to m} - \mathbf{t}^1_{1\to l} \tag{18}$$

$$\mathbf{q}_k = R^T_{1\to k}K_k^{-1}\mathbf{p}_i^k \tag{19}$$

$$\mathbf{q}_l = R^T_{1\to l}K_l^{-1}\mathbf{p}_i^l \tag{20}$$

$$\mathbf{q}_m = R^T_{1\to m}K_m^{-1}\mathbf{p}_i^m \tag{21}$$

where $R_{1\to s}$, $s \in \{k,l,m\}$, is a rotation matrix computed using $\mathbf{\Psi}_{1\to s}$, that is part of the state vector $\mathbf{x}^{rel}$.

Denote the residual error for the $i$th observed landmark ($i \in \{1,\ldots,M\}$) by

$$\mathbf{z}_i^{(k,l,m)} \triangleq \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T. \tag{22}$$

This residual can therefore be written as

$$\mathbf{z}_i^{(k,l,m)} = \mathbf{h}_i^{(k,l,m)}(\hat{\mathbf{x}}_{1\to k}, \hat{\mathbf{x}}_{1\to l}, \hat{\mathbf{x}}_{1\to m}, \mathbf{p}_i^k, \mathbf{p}_i^l, \mathbf{p}_i^m) \tag{23}$$

and thus

$$\mathbf{z}_i^{(k,l,m)} = \mathbf{h}_i^{(k,l,m)}(\hat{\mathbf{x}}^{rel}, \mathbf{p}_i^k, \mathbf{p}_i^l, \mathbf{p}_i^m) \tag{24}$$

where $\mathbf{h}_i^{(k,l,m)}$ represents the nonlinear, known, function given by the three-view constraints (19)-(21). In Eqs. (23)-(24), $\hat{\mathbf{a}}$ denotes the estimation of some vector $\mathbf{a}$.

### C. General Observations of Landmarks

So far, only three-view correspondences were considered. In practice, landmarks can be observed in any number of views. Consider the $i$th landmark $\mathbf{P}_i$ is observed in $n_i \in \{1,\ldots,N\}$ views.

When the landmark is observed by less than three views, it is only possible to apply the epipolar geometry constraint (14) in case of a two-view observation ($n_i = 2$). No multi-view constraints can be formulated at all if the landmark is only observed in a single view.

The more interesting case, however, occurs when $n_i > 3$. One possible alternative is to apply the three-view constraints (14)-(16) on three views chosen from the $n_i$ views. Different reasoning can be applied when choosing these three views. The drawback of such an alternative is that not all the available information is actually used, since the other $n_i - 3$ views do not participate in the constraints.

In order to exploit all the available information in the $n_i$ corresponding points, the sequence of $n_i$ views $\{k_j\}_{j=1}^{n_i}$ can be split into sequential, overlapping, triplets as follows:

$$\{(k_1,k_2,k_3),(k_2,k_3,k_4),\ldots,(k_{n_i-2},k_{n_i-1},k_{n_i})\} \tag{25}$$

For example, the first triplet would be composed of views $k_1, k_2$ and $k_3$, while the next triplet would be views $k_2, k_3$ and $k_4$.

To avoid double-counting, only *independent* three-view constraints along the given sequence should be applied. In case of the previous example, applying the three-view constraints (14)-(16) would yield the same epipolar constraint between views $k_2$ and $k_3$ twice: once as the constraint (15) when processing the views $k_1, k_2$ and $k_3$, and once as the constraint (14) while processing the views $k_2, k_3$ and $k_4$.

Consequently, to avoid using the same measurements (and constraints) more than once, after formulating the three-view constraints (14)-(16) for the first three views $k_1, k_2$ and $k_3$, only two constraints (15) and (16) are applied for the following $n_i - 3$ triplets of views. Denote by $\mathbf{z}_i^{(klm)*}$ the residuals in this reduced version:

$$\mathbf{z}_i^{(k,l,m)*} \triangleq \begin{bmatrix} z_2 & z_3 \end{bmatrix}^T$$

Stacking all the constraints that can be written for the $n_i$ views that observe the $i$th landmark yields

$$\mathbf{z}_i \triangleq \begin{bmatrix} \mathbf{z}_i^{(k_1k_2k_3)} \\ \mathbf{z}_i^{(k_2,k_3,k_4)*} \\ \vdots \\ \mathbf{z}_i^{(k_{n_i-2},k_{n_i-1},k_{n_i})*} \end{bmatrix}$$

Since only the first triplet contributes all the three-view constraints (14)-(16), while each of the $n_i - 3$ other

Table I
CONSIDERED SCENARIO IN A BASIC EXAMPLE

| | View 1 | View 2 | View 3 | View 4 | View 5 |
|---|---|---|---|---|---|
| 3D point #1 | × | | × | × | |
| 3D point #2 | × | × | | × | × |
| 3D point #3 | | | × | × | × |

triplets contribute two constraints, the dimensions of $\mathbf{z}_i$ are $(2n_i - 3) \times 1$.

Considering *all* the observed 3D points $\mathbf{P}_i$ with $i = 1, \ldots, M$, the overall residual vector $\mathbf{z}$ can be written for the given sequence of $N$ views and the $M$ observed 3D points as:

$$\mathbf{z} \triangleq \begin{bmatrix} \mathbf{z}_1^T & \ldots & \mathbf{z}_M^T \end{bmatrix} = \mathbf{h}\left(\hat{\mathbf{x}}^{rel}, \mathbf{y}\right)$$

where $\mathbf{y}$ is a vector of all the pixel observations.

The multi-view constraints function $\mathbf{h}$ is part of the cost function $J^{LBA}$, defined in Eqs. (4) and (7).

### D. A Basic Example

Consider a scenario of 5 views observing 3 3D points ($N = 5, M = 3$), as shown in Table I. As seen, the first and third 3D points ($\mathbf{P}_1$ and $\mathbf{P}_3$) are observed in 3 images: views $(1, 3, 4)$ and views $(3, 4, 5)$, respectively. The second 3D point ($\mathbf{P}_2$) is observed in all views except the second view.

Thus, $n_1 = n_3 = 3$ and $n_2 = 4$, while the sequence of views $\left\{k_j^i\right\}_{i=1}^{n_i}$ for each landmark is

$$\{k_j^1\}_{j=1}^{n_{1=3}} = \{1, 3, 4\} \quad (26)$$

$$\{k_j^2\}_{j=1}^{n_2=4} = \{1, 2, 4, 5\} \quad (27)$$

$$\{k_j^3\}_{j=1}^{n_3=3} = \{3, 4, 5\} \quad (28)$$

Since the second landmark is observed by more than three views, the following sequential overlapping triplets are used when applying the three-view constraints (cf. Eq. (25)):$\{(1, 2, 4), (2, 4, 5)\}$.

Consequently, the residual vectors $\mathbf{z}_i$ are

$$\mathbf{z}_1 = \mathbf{z}_1^{(1,3,4)} \in \mathbb{R}^{3\times 1} \quad (29)$$

$$\mathbf{z}_2 = \begin{bmatrix} \mathbf{z}_2^{(1,2,4)} \\ \mathbf{z}_2^{(2,4,5)*} \end{bmatrix} \in \mathbb{R}^{5\times 1} \quad (30)$$

$$\mathbf{z}_3 = \mathbf{z}_3^{(3,4,5)} \in \mathbb{R}^{3\times 1} \quad (31)$$

and the overall residual $\mathbf{z}$ is

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1^T & \mathbf{z}_2^T & \mathbf{z}_3^T \end{bmatrix}^T \in \mathbb{R}^{11\times 1} \quad (32)$$

### E. Rank Deficiency and an Additional Scale Constraint

It is well known, that relying purely on imagery information and assuming calibrated cameras, it is possible to estimate the camera matrices and to perform structure reconstruction only up to a similarity transformation. This similarity transformation represents a 7 degree-of-freedom (DOF) ambiguity [4], [17]: translation and rotation of the reference system and the overall scale of the observed scene. Thus, the resulting set of equations is rank-deficient, with 7 zero singular values.

While the relative formulation (cf. Section IV-A) fixes the translation and rotation DOFs, the overall scale is not determined: The three-view constraints (8)-(10) allow estimating the camera matrices in the whole sequence with a consistent, but unknown, scale. Therefore, the system remains to be rank-deficient and should be handled with a proper regularization.

One possible alternative for performing a regularization, is to impose an additional set of constraints $\mathbf{g}\left(\mathbf{x}^{rel}\right)$ on the system, which in the general case, are non-linear. This set of constraints can be introduced to the cost function $J^{LBA}$, defined in Eq. (4), either by augmentation to the multi-view constraints $\mathbf{h}$, or by a new Lagrange multipliers vector. In the latter case, the new cost function would be

$$J^{LBA} = \|\mathbf{p} - \hat{\mathbf{p}}\|_{\Sigma}^2 - 2\boldsymbol{\lambda}_1^T \mathbf{h}\left(\hat{\mathbf{x}}^{rel}, \hat{\mathbf{p}}\right) - 2\lambda_2^T \mathbf{g}\left(\hat{\mathbf{x}}^{rel}\right)$$

In certain applications, it makes sense to explicitly specify the scale parameter, thereby significantly improving the condition number of the system. For example, in the context of inertial navigation, it is reasonable to assume that the navigation solution at the first few moments is relatively accurate (compared to the navigation accuracy after some time) since the inertial navigation errors have not yet become significant. Consequently, the scale parameter can be determined, with the proper uncertainty, based on the navigation information from these first moments. In particular, it is possible to calculate the translation vector, including the magnitude, between the first two views in the sequence, thus implicitly setting the scale of the problem. In such cases, the resulting scale constraint will be linear.

## V. STRUCTURE RECONSTRUCTION

After estimating the relative state vector $\mathbf{x}^{rel}$, it is possible to perform structure reconstruction. This is as opposed to the conventional BA, in which structure reconstruction and pose estimation are performed simultaneously.

The projection matrix relating between the unknown 3D point coordinates $\mathbf{P}$, expressed relative to the first view, and the image coordinate $\mathbf{p}^j$ of this 3D point in some view $j \in \{1, \ldots, N\}$ is:

$$\mathbf{p}_j = M_j \mathbf{P}$$

with the projection matrix $M_j$ defined in Eq. (13):

$$M_j = K_j \begin{bmatrix} R_{1\to j} & \mathbf{t}^j_{j\to 1} \end{bmatrix}.$$

This matrix can be calculated based on the optimized values of $\mathbf{x}^{rel}$.

The actual structure reconstruction process can be carried out using a standard procedure [4]. Each view $j$ contributes two independent equations of the form

$$A_j \tilde{\mathbf{P}} = \mathbf{b}_j \tag{33}$$

with $\tilde{\mathbf{P}}$ denoting inhomogeneous coordinates of the 3D point. These equations can be written in a homogeneous form as:

$$\begin{bmatrix} u_j M_j^{3T} - M_j^{1T} \\ v_j M_j^{3T} - M_j^{2T} \end{bmatrix} \mathbf{P} = \mathbf{0} \tag{34}$$

where $M_j^{uT}$ indicates the $u$th row of $M$. Stacking the equations together from all the relevant views, yields a homogeneous over-determined system of equations, which can be solved using non-iterative least-squares (e.g., using SVD).

However, in practical applications, the camera matrices and the image features are imperfect and accompanied with uncertainty covariance matrices. In particular, the optimized camera poses, represented by $\mathbf{x}^{rel}$, are accompanied with an a posteriori covariance matrix, calculated during the optimization process (cf. Appendix). Such information can be used as regularization terms, as described in the Appendix, in the structure estimation process.

To this end, the inhomogeneous form of the reprojection equation (33) is used. Stacking equations (33) for all the participating views $j$ together yields:

$$A \tilde{\mathbf{P}} = \mathbf{b} \tag{35}$$

Applying the optimization on the above equation allows to recover the landmark $\mathbf{P}$ and the uncertainty covariance, while using the a posteriori covariance of $\mathbf{x}^{rel}$ as a regularization term.

## VI. Results

In this section the developed approach is demonstrated on the on-line available Pozzoveggiani dataset[1] [2]. The dataset contains real imagery and a ground truth data for the camera pose and the observed 3D points. Figure 1 shows several images from the dataset, while Figure 2 shows the 3D points cloud and the camera locations (only part of the dataset is shown). The proposed approach for light bundle adjustment was applied on a set of 42 images out of the 48 images in the dataset.

Initial values of the relative state vector $\mathbf{x}^{rel}$ were obtained by contaminating the ground truth data as follows. Position and attitude errors were drawn from a zero-mean Gaussian distribution with a standard deviation of $50$ meters and $0.1$ degrees, respectively, in each axis. These errors were then used for corrupting the camera pose ground truth of all the views in the sequence.

The assumed standard deviation for initial position errors is *very large* for the considered scenario, as the location of all cameras in the dataset can be bounded by a $70 \times 70$ meters area (cf. Figure 2). These errors were drawn independently for each view, as if the images were taken by different users or captured by independently moving robots (or any other uncorrelated sources).

Since measured image pixels are not included in the dataset, ideal pixel coordinates were calculated by projecting the ground truth 3D points using ground truth camera pose data. The measured image pixels were taken as the ideal image pixels, corrupted with a Gaussian zero-mean noise with a 0.5 pixel standard deviation.

As described in Section IV-E, an additional scale constraint as a regularization term. In the current implementation, this scale constraint was obtained by assuming a perfect translation between the first two views in the sequence.

Figure 3 shows the estimated camera position (relative to the first view) for all the $42$ views in the sequence, compared to the ground truth values and to the initial camera pose values (that were used as initial solution in the optimization process). As can be observed, the a posteriori camera position is very improved. The actual position estimation errors are shown in Figure 4(b): a priori errors are on the order of $50$ meters, in each axis, errors after the optimization are reduced to a few meters.

Structure estimation was performed using initial camera pose values and based on the optimized camera pose values, as described in Section V. Figure 5 shows the structure estimation errors in each case. Errors after

[1] http://profs.sci.univr.it/~fusiello/demo/samantha/

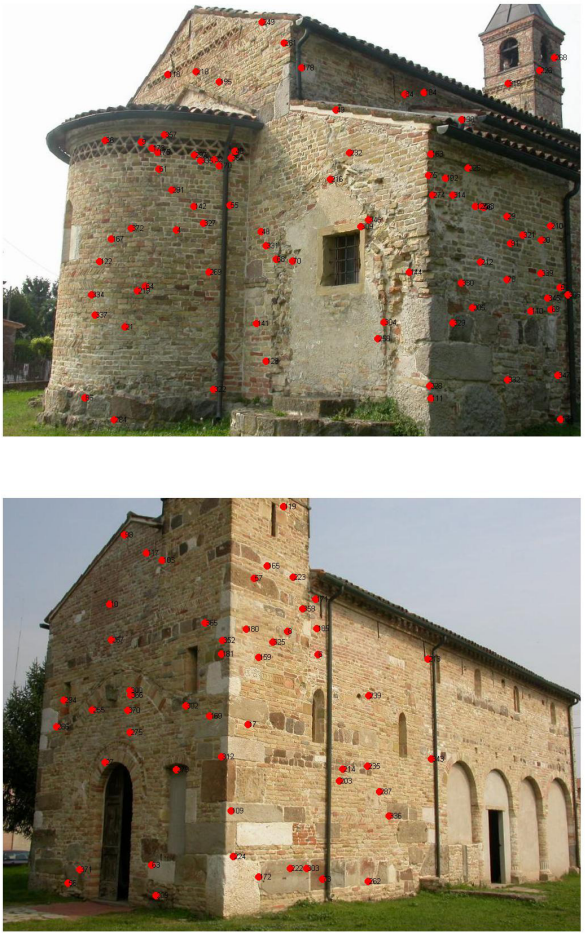

Figure 1. Several images from the Pozzoveggiani dataset. Red markings indicate ground truth features (see text).

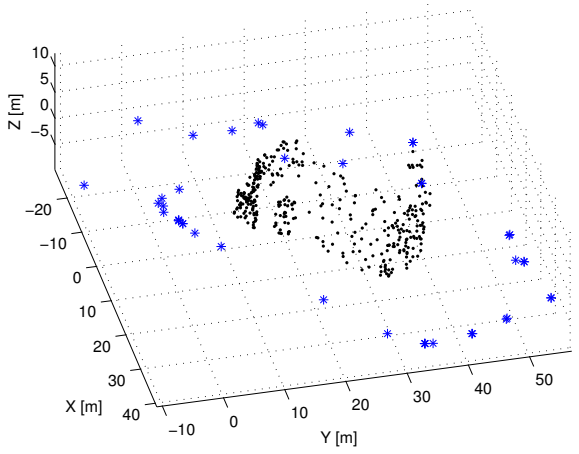


Figure 2. 3D points cloud and camera locations in the Pozzoveggiani dataset (only part of the data is shown). Blue star markings represent camera locations, 3D points are indicated by black dot markings.

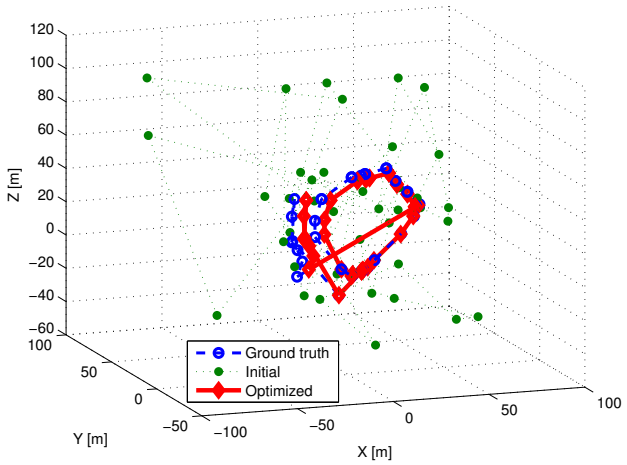


Figure 3. Optimized position compared to the ground truth and to the initial position values.

the optimization are significantly reduced, with typical values on the order of one meters in each axis (with a few exceptions). A priori errors are on the order of $100-200$ meters in each axis, with several extremely large errors obtained for landmarks that are observed from only a few cameras.

## VII. Conclusions

This paper presented a new approach for bundle adjustment, in which the non-linear optimization process involved only the camera poses in a given sequence of views. While structure variables are not part of the optimization, structure reconstruction can be performed based on the optimized camera poses. As opposed to the projection equations that are used in conventional bundle adjustment, the cost function in the proposed approach was formulated using a recently developed formulation for three-view geometry constraints. The reduced number of the optimized variables yielded an improvement in the computational complexity of the optimization process. Preliminary results were provided, demonstrating the proposed method on a publicly available dataset of real images.

## Appendix

This appendix describes the optimization process for the cost function $J^{LBA}$, defined in Eqs. (4) and (7). Denote by $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{p}}_k$ the corrected camera poses and fitted measurements obtained at the $k$th iteration, and by $\Delta\mathbf{x}_k$ and $\Delta\mathbf{p}_k$ the actual corrections:

$$\begin{aligned}\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} + \Delta\mathbf{x}_k \\ \hat{\mathbf{p}}_k &= \hat{\mathbf{p}}_{k-1} + \Delta\mathbf{p}_k\end{aligned} \tag{36}$$

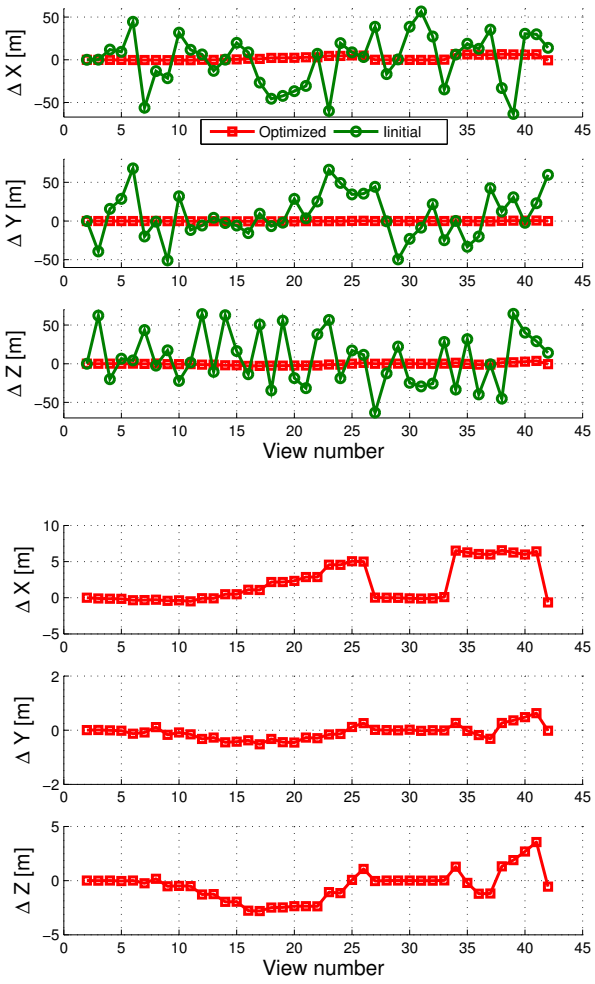


Figure 4. (top) Position errors before and after optimization. (bottom) Zoom on position errors after optimization.

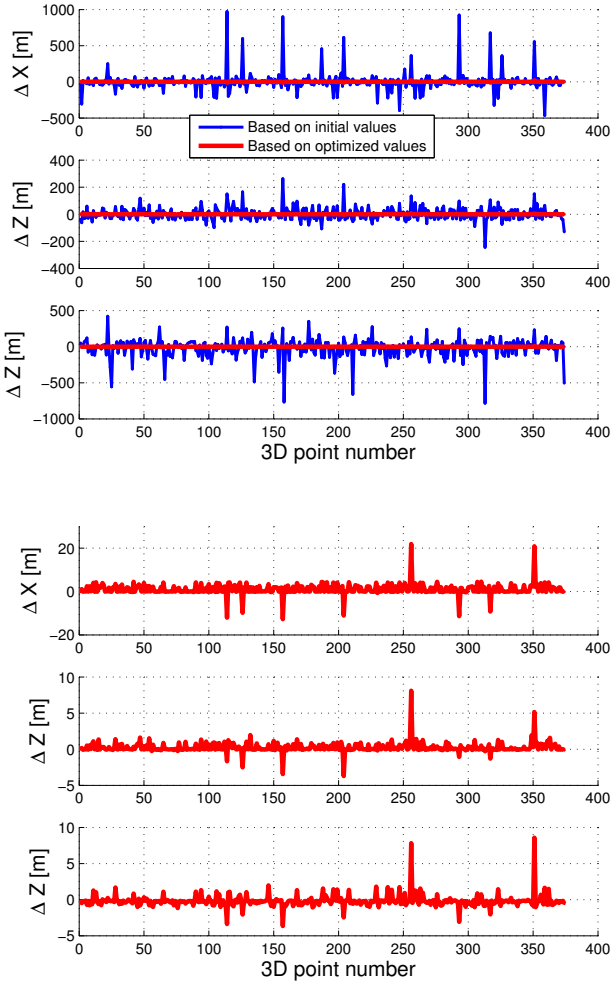


Figure 5. Structure reconstruction errors. Typical precision after light bundle adjustment is around 1 meter. Estimation errors of individual few points are larger. (top) Comparison between structure reconstruction based on initial camera pose values and camera pose after the optimization. (bottom) Zoom on estimation errors based on optimized camera pose values.

with $\hat{\mathbf{x}}_0$ being the initial value of $\mathbf{x}$ and $\hat{\mathbf{p}}_0 \equiv \mathbf{p}$. Consider minimizing this cost function at the $k$th iteration:

$$J_k^{LBA} \triangleq \|\mathbf{p} - \hat{\mathbf{p}}_k\|_\Sigma^2 - 2\boldsymbol{\lambda}^T \mathbf{h}(\hat{\mathbf{x}}_k, \hat{\mathbf{p}}_k) \tag{37}$$

The goal at the $k$th iteration is to find $\Delta\mathbf{x}_k$ and $\Delta\mathbf{p}_k$ that minimize $J_k^{LBA}$. Denoting $v = \mathbf{p} - \hat{\mathbf{p}}_k$ and linearizing the constraints function $\mathbf{h}$ gives

$$J_k^{LBA} = \|\mathbf{v}\|_\Sigma^2 - 2\boldsymbol{\lambda}^T \left(\mathbf{z}_k + A_k \Delta\mathbf{x}_k + B_k \mathbf{v}\right)$$

where $A_k, B_k$ are the Jacobian matrices

$$A_k \triangleq \nabla_{\mathbf{x}} \mathbf{h} \quad , \quad B_k \triangleq \nabla_{\mathbf{p}} \mathbf{h} \tag{38}$$

evaluated about $\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{p}}_{k-1}$ and

$$\mathbf{z}_k \triangleq \mathbf{h}(\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{p}}_{k-1})$$

Taking derivatives of $J_k^{LBA}$ with respect to $\mathbf{p}_k, \mathbf{x}_k$ and $\boldsymbol{\lambda}$ produces the following expressions for $\Delta\mathbf{p}_k, \Delta\mathbf{x}_k$:

$$\Delta\mathbf{x}_k = -\left(A_k^T M^{-1} A_k\right)^{-1} A_k^T M^{-1} \mathbf{z}_k \tag{39}$$

$$\Delta\mathbf{p}_k = -\Sigma B_k^T M^{-1} \left(\mathbf{z}_k + A_k \Delta\mathbf{x}_k\right) \tag{40}$$

where $M \triangleq B_k \Sigma B_k^T$. After convergence of the iterations, it is possible to calculate the a posteriori covariance $P_+$ as

$$P_+ = \frac{\mathbf{v}^T \Sigma^{-1} \mathbf{v}}{n_y - n_x} \left(A_k^T M^{-1} A_k\right)^{-1} \tag{41}$$

where $n_x$ is the number of parameters in $\mathbf{x}$, and $n_y$ is the number of elements in $\mathbf{y}$: $\mathbf{x} \in \mathbb{R}^{n_x \times 1}, \mathbf{y} \in \mathbb{R}^{n_y \times 1}$.

If the initial solution for camera poses is accompanied with an uncertainty covariance $P_0$, as common in robotics navigation applications, it is possible to introduce a regularization term into the cost function $J^{LBA}$:

$$J_k^{LBA} = \|\mathbf{v}\|_{\Sigma}^2 + \|\Delta\mathbf{x}_0\|_{P_0}^2 - 2\boldsymbol{\lambda}^T\mathbf{h}(\hat{\mathbf{x}}_k, \mathbf{p})$$

Bringing this cost function to a minimum, and noting the relation $\Delta\mathbf{x}_0 = \Delta\mathbf{x}_k + \hat{\mathbf{x}}_{k-1} - \hat{\mathbf{x}}_0$, the expression for $\Delta\mathbf{x}_k$ changes from Eq. (39) into:

$$\Delta\mathbf{x}_k = -\left(A_k^T M^{-1} A_k + P_0^{-1}\right)^{-1}\mathbf{u}$$

with

$$\mathbf{u} \triangleq P_0^{-1}\left(\hat{\mathbf{x}}_{k-1} - \hat{\mathbf{x}}_0\right) + A_k^T M^{-1}\mathbf{z}_k.$$

In addition, the expression for calculating the a posteriori covariance $P_+$ changes (from Eq. (41)) to

$$P_+ = \frac{\mathbf{v}^T\Sigma^{-1}\mathbf{v}}{n_y - n_x}\left(A_k^T M^{-1} A_k + P_0^{-1}\right)^{-1}$$

while $\Delta\mathbf{p}_k$ is calculated without any change (cf. Eq. (40)).

## REFERENCES


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

[2] A. M. Farenzena, A. Fusiello, and R. Gherardi. Structure-and-motion pipeline on a hierarchical cluster tree. *Proceedings of the IEEE International Workshop on 3-D Digital Imaging and Modeling*, pages 1489 – 1496, October 2009.

[3] A. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proceedings of the European Conference on Computer Vision*, pages 311–326, 1998.

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

[5] V. Indelman. Navigation performance enhancement using online mosaicking. Technion, Israel, 2011.

[6] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein. Real-time vision-aided localization and navigation based on three-view geometry. *IEEE Transactions on Aerospace and Electronic Systems*, 48(2), 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(2), 2012.

[8] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein. Distributed vision-aided cooperative localization and navigation based on three-view geometry. *Robotics and Autonomous Systems*, 2012, to appear.

[9] K. Konolige and M. Agrawal. Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, 2008.

[10] B. Liu, M. Yu, D. Maier, and R. Manner. Accelerated bundle adjustment in multiple-view reconstruction. *Lecture Notes in Computer Science*, 2774:1195–1201, 2003.

[11] B. Liu, M. Yu, D. Maier, and R. Männer. An efficient and accurate method for 3d-point reconstruction from multiple views. *Int. J. Comput. Vision*, 65(3):175–188, Dec. 2005.

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

[13] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. 3d reconstruction of complex structures with bundle adjustment: an incremental approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

[14] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(6):756–770, June 2004.

[15] G. Sibley, C. Mei, I. Reid, and P. Newman. Adaptive relative bundle adjustment. In *Robotics Science and Systems (RSS)*, Seattle, USA, June 2009.

[16] R. Steffen, J.-m. Frahm, and W. F. Relative bundle adjustment based on trifocal constraints. *ECCV Workshop on Reconstruction and Modeling of Large-Scale 3D Virtual Environments*, 2010.

[17] H. Strasdat, J. M. M. Montiel, and A. J. Davison. Scale drift-aware large scale monocular SLAM. Zaragoza, Spain, June 2010.

[18] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Vision Algorithms: Theory and Practice, LNCS*, pages 298–375. Springer Verlag, 2000.

[19] R. Vidal, Y. Ma, S. Hsu, and S. Sastry. Optimal motion estimation from multiview normalized epipolar constraint. *ICCV*, 1:34–41, 2001.

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