ABSTRACT:

The global approaches solve SfM problems by independently inferring relative motions, followed by a sequential estimation of global rotations and translations. It is a fast approach but not optimal because it relies only on pairs and triplets of images and it is not a joint optimisation. In this publication we present a methodology that increases the quality of global solutions without the usual computational burden tied to the bundle adjustment. We propose an efficient structure approximation approach that relies on relative motions known upfront. Using the approximated structure, we are capable of refining the initial poses at very low computational cost. Compared to different benchmark datasets and software solutions, our approach improves the processing times while maintaining good accuracy.

1. INTRODUCTION

Photogrammetry and computer vision Structure from Motion (SfM) algorithms underwent a remarkable evolution during the past two decades. Much of its evolution was driven by the advances in sensor technology, growing computer capabilities and democratisation of the fields through different software solutions (Snavely et al., 2006; Pierrot Desesligny and Cléry, 2011; Moulon et al., 2016; Schönberger and Frahm, 2016). Existing SfM approaches for pose estimation become computationally inefficient as the number of images increases. Two solutions exist: sequential and global methods. Sequential methods (Schönberger and Frahm, 2015; Snavely et al., 2006) generate one or multiple (e.g. hierarchical methods (Klopschitz et al., 2010; Toldo et al., 2015)) seed images and sequentially concatenate overlapping images. Non-linear least squares bundle adjustment is then used to systematically eliminate accumulated errors (Triggs et al., 1999; Forstner and Wöbbel, 2016). On the other hand, global methods (Govindu, 2001; Moulon et al., 2013; Wilson and Snavely, 2014) simultaneously and independently find relationships between pairs or triplets of images. These relationships are encoded in an epipolar graph which forms the basis of inferring the global rotations and translations. Global methods require the knowledge of image correspondences only in the first phase when computing the two-view or three-view geometries. This makes them faster and more computationally efficient. However, identifying outliers becomes challenging because the computation is bound to a pair or a triplet.

Global solutions to SfM problems provide only initialisation of camera poses. Ordinarily, a final bundle adjustment (BA) including poses and point correspondences is run to calculate the optimal solution (Triggs et al., 1999), which is a costly processing. The objective of this work is to go towards truly structureless bundle adjustment by refining the initial poses with very few points. We propose a structure (i.e. 3D points) approximation algorithm that relies on pairs and triplets of images. We firstly introduce a robust method for the relative motion calculation. Then, we redefine the per-relative-motion structure through the intermediary of an ellipsoid. The goal here is to generate a minimal set of 3D points that are sufficient to infer the local structure. This set is later used to refine the global solution in a bundle adjustment routine. Please note that we do not present a global motion approach in this work. Instead, to produce an initialization to the BA, we use the global approach of SfmInit (Wilson and Snavely, 2014) or MicMac (Pierrot Desesligny and Cléry, 2011; Rupnik et al., 2017). We evaluate our approach on the Strecha benchmark (Strecha et al., 2008), as well as sequential acquisitions by a mobile mapping platform and a drone. All methods are implemented and available from: github.com/micmacIGN. The sequential datasets and their respective ground truth are also available upon request.

2. RELATED WORK

On global motion methods

Determination of global motions is performed separately for the rotational and translation parts (Govindu, 2001). To calculate the rotations, the existing solutions: ignore the orthogonality constraint and employ linear methods (Martinec and Pajdla, 2007; Arie-Nachimson et al., 2012; Moulon et al., 2013); or optimise with robust $\ell_1$ as Hartley et al. (Hartley et al., 2011) or as Chatterjee et al. (Chatterjee and Govindu, 2013) who base their algorithm on the Lie Group. A review of rotation averaging algorithms was published by Hartley et al. (Hartley et al., 2013). The translation component is always computed in the second place. Unlike for the rotational component, there is no direct way to compute scale-consistent translations from a set of available pairwise constraints. Some methods exploit only pairwise constraints (Govindu, 2001; Martinec and Pajdla, 2007; Arie-Nachimson et al., 2012; Wilson and Snavely, 2014), but are susceptible to a degenerate solution when all cameras are aligned. Other methods formulate the problem using trifocal dependencies (Courchay et al., 2010; Zach et al., 2010; Jiang et al., 2013; Moulon et al., 2013). Moreover, approaches based on linear solvers (Arie-Nachimson et al., 2012; Govindu, 2001), and $\ell_\infty$ norm optimisation (Sim and Hartley, 2006; Kahl and Hartley, 2008; Moulon et al., 2013) can be distinguished. Methods formulating the BA estimation problem in terms of epipolar constraints (Steffen et al., 2013) are also highlighted.
On outliers in relative motions Relative motions are the main ingredient in global pose estimation methods. This optimization scheme is particularly sensitive to outliers due to a limited number of motion observations. Since the motions are derived from automated algorithms for sparse correspondence search (e.g. SIFT [Lowe, 2004]), they inherently contain outliers, especially across ambiguous scenes with repetitive patterns, low texture, moving objects, etc. Existing approaches cope with the outliers by either eliminating them prior to optimising for global rotations and translations (i.e. filtering), or by adopting robust optimisation techniques. Examples of the latter are robust $\ell_1$ optimization used for rotation averaging [Hartley et al., 2011] [Chatterjee and Govindu, 2013] and translations solving [Dalalany and Keriven, 2009] [Olsson et al., 2010] [Zhu et al., 2017]. Some recent works have also addressed the evident complementarity of sequential and global methods. In Zhou et al. [Zhu et al., 2017] [Zhu et al., 2018] many local incremental SFMs help to discard mismatches and invalid motions. Among the solutions based on filters, the pioneering works [Govindu, 2006] [Sim and Hartley, 2006] [Martinec and Pajdla, 2007] focus on pairwise constraints and remove the outliers by: RANSAC epipolar sampling [Govindu, 2006]; using iterative removal techniques [Sim and Hartley, 2006]; or identifying false matches via partial reconstructions [Martinec and Pajdla, 2007]. Later works exploit the redundancy in the epipolar graph by looking at triplets of images. Courchay et al. [Courchay et al., 2010] introduce a trifocal graph parametrization and estimate robust global poses using linear programming with imposed loop constraints and RANSAC. Similarly to this work, Zach et al. [Zach et al., 2010] uses loopy belief propagation to identify erroneous geometries in loops. Jiang et al. [Jiang et al., 2013] establishes a trifocal coplanarity constraint that minimizes a geometric error. Work in Moulon et al. [Moulon et al., 2013] fuses the Bayesian inference of Zach et al. [Zach et al., 2010] and weighted graph filtering of Enqvist et al. [Enqvist et al., 2011] to remove outliers. Sweeney et al. [Sweeney et al., 2015] updates the fundamental matrices by optimizing a cost function defined over the trifocal point transfer. Finally, Wilson et al. [Wilson and Snavely, 2014] finds a purely pairwise method adapted to very large image collection where pairwise translations are projected to 1D subspace and a global ordering fitting the pairwise constraints is sought.

CONTRIBUTION AND OVERVIEW

To increase the robustness of the relative motions, our method combines several hypotheses, and tests each with a suitable direct algorithm (Section 3.1). To infer the triplets we take a similar perspective as Jiang et al., 2013 and Sweeney et al., 2015. We present a linear algorithm that rapidly calculates the relative motions (Quick triplet algorithm). The robust triplet algorithm (Section 3.2) tests two hypotheses to see whether the camera in question has a short, normal, or a long focal length. As the long focal lengths approach the orthogonal projection, we replace the perspective camera model with the one proposed by Tomasi et al. [Tomasi and Kanade, 1992]. Otherwise, the pose of the “third” image of the triplet is found with the spatial resection [Grunert, 1841]. We intentionally avoided the trifocal tensor parametrization knowing its unstable behavior on flat surfaces [Julià and Monasse, 2017].

Our structure reduction approach (Section 4) approximates the original points with an ellipsoid, and generates a small set of virtual points within its volume. This reduction allows for a lighter BA, without having to resort to PCG-based optimizations [Agarwal et al., 2010] [Jeong et al., 2011] [Kushal and Agarwal, 2012]. Therefore, we keep the access to the reduced camera system’s covariance matrix. This is advantageous for many metrological applications that necessitate a degree of confidence associated with their measurements.

3. BUILDING THE RELATIVE MOTIONS

The computation of relative motion consistent within triplets of images is divided into three stages. Firstly, given a set of image correspondences, we create the epipolar hypergraph $\mathcal{H}(V, E)$, where each vertex $V_i \in V$ represents an image, and the edge $E_{ij}$ connects a pair of images $(V_i, V_j)$ (Section 3.1). The connection implies that relative motion $\{R_{ij}, t_{ij}\}$ is known. We refer to $\mathcal{H}(V, E)$ as a hypergraph because we have access to a list of connected vertices $V_i$ for a given $E_{ij}$. The connected vertices are then explored to construct triplets complying with the respective edges (Section 3.2). We assume that cameras are calibrated, i.e. at least their focal lengths and the principal points are known. The local coordinate frame within an edge $E_{ij}$ or a triplet $(V_i, V_j, V_k)$ is always associated with the first camera (i.e. $V_i$). In the text we often refer to the “third” image of a triplet by which we mean the image associated with $V_k$.

3.1 Pairwise relative motions

Several direct algorithms are tested to compute robust relative motions between pairs of images. We begin with the hypothesis that the corresponding points contain a many outliers, and slowly relax the constraint by allowing more points in the computation. The full set of correspondences entering the method amounts to 500 and was chosen randomly. In total we test 6 calculation variants, see Table 1. The difference between variant T1 and T2 is in the way the points a drawn out the set of 500 points. While T2 is purely random, in T1 we bias the selection by forcing that the points are well distributed across the image plane. This way we avoid sampling points that are clustered in one zone. Variants T1-T4 are solved with 8-point algorithm [Hartley and Zisserman, 2003] throughout several RANSAC samples. After each draw the current solution is refined with a $t_2$ solver. In T5 we employ the full set of points, and perform $t_1$ and $t_2$ estimations. The T6 variant tests the planarity of the scene by computing a homography and decomposing it to a relative motion [Faugeras and Lustman, 1988]. Finally, the variant with the smallest re-projection error is refined in a coplanarity-based bundle adjustment.

| Variant | $N_{pts}$ | $N_{RANSAC}$ | $t_1 / t_2$ | Model |
|---------|-----------|-------------|-------------|-------|
| T1      | 8*        | 200         | $t_2$       | $M_{8x8}$ |
| T2      | 8         | 50          | $t_2$       | $M_{8x8}$ |
| T3      | 12        | 100         | $t_2$       | $M_{8x8}$ |
| T4      | 250       | 20          | $t_2$       | $M_{8x8}$ |
| T5      | 500       | 0           | $t_1$ & $t_2$ | $M_{8x8}$ |
| T6      | 130       | 20          | $t_2$       | H      |

Table 1. Six variants to calculate the pairwise relative motions. $M_{8x8}$ is the 8-point essential matrix algorithm, H is the homography decomposed to a relative motion [Faugeras and Lustman, 1988]. (*) indicates that the points were drawn with a bias forcing a homogenous distribution within the image.
3.2 Triplets motions

The triplet motions are determined ideally between "relevant" triplets of images. In order to select a subset of suitable triplets, we first determine the triplet motions between all possible candidates (see Quick triplet algorithm and Algorithm[1]), and use them to define a per-triplet quality index (see Triplet selection). The final triplet motions, calculated on a pre-selected set, result from the Robust triplet algorithm.

On straight lines intersection For the sake of clarity, we lay out the line intersection algorithm that is the building block of the Quick triplet algorithm. Let us define two straight lines as (cf. Figure[1](a)): \( P = P_0+p(P_1-P_0), Q = Q_0+q(Q_1-Q_0) \). The intersecting 3D point belongs to the line that minimizes the distance between the lines \( P \) and \( Q \), hence, it will be found on a line perpendicular to them. We define it as \((P-Q)\), and impose the following: \((P-Q)(P_1-P_0) = 0, (P-Q)(Q_1-Q_0) = 0\). Using the above, parameters \( p \) and \( q \) are inferred. The final 3D point is equal to \( \frac{1}{2} \left( P_{int_p} + P_{int_q} \right) \), where \( P_{int_p} = P_0 + (P_1-P_0) \cdot p \), \( P_{int_q} = Q_0 + (Q_1-Q_0) \cdot q \).

Quick triplet algorithm The rotation of the "third" image in the triplet \((V_i, V_j, V_k)\) is calculated twice: directly as \( R'_i = R_{31} \), and indirectly as \( R''_i = R_{32} \cdot R_{33} \). The definite rotation being \( R_i = \frac{1}{2} (R'_i + R''_i) \). The orthogonality of \( R_i \) is enforced by mapping it to its nearest orthogonal rotation with Singular Value Decomposition.

The computation of the perspective center proceeds in two steps as presented in Figure[1] and Algorithm[1]. Our objective is to force the perspective center to lie close to both the translation direction \( \mathbf{t}_{31} \) calculated in the preceding step, and the directions associated with the position of 3D points. To achieve this, our first step is to determine the 3D position of two corresponding points using image measurements in \( V_i \) and \( V_j \) (cf. Figure[1](b)). Then, the new perspective center of \( V_k \) results from intersecting the translation direction \( \mathbf{t}_{31} \) and the direction \( \mathbf{t}_{3P_{int}} \), defined by the image measurement in \( V_k \) and attached at the position of the 3D point. For \( N \) corresponding points, one obtains \( N \) estimates of the perspective center (or \( N \) estimates of \( p \) and \( q \), see Figure[1](a)). The terminal value is then \( C_3 = \mathbf{t}_{31} \cdot \mathbf{p}, \) where \( \mathbf{p} = \left\{ P_i \right\}, \mathbf{p} \) is the median of \( p \) and \( i \in [1, N] \). By definition, the proposed algorithm calculates consistent triplets exclusively from image correspondences of manifold 3 (i.e., points visible in 3 images). Statistically speaking, higher manifold points are less susceptible to outliers, therefore they are expected to produce more stable results. It is also required that a minimum of 8 manifold points exists. In practice, many more points are available and we decimate them by a factor of 50 when the number exceeds 500. Note that in this formulation, the computation of the perspective center is not degenerate if all three perspective centers are located on a straight line.

Triplet selection Out of a number of initialized triplets, we want to select the \( K \) best ones. The notion of a best triplet is translated to a quality index \( Q \) and calculated for each triplet in \( \mathcal{H} \). The \( Q \) favours wide baselines between images (i.e. base-to-height ratios or \( bh \)) as presented here:

\[
R = \frac{bh}{bh + TL},
\]

\[
Q = min(R_{V_iV_k}, R_{V_jV_k}), \quad (1)
\]

where \( TL \) is a rough \( bh \) limit; \( R_{V_iV_k} \) is computed between the first and the second image in a triplet, and \( R_{V_jV_k} \) between the second and the third, correspondingly. To rapidly calculate the indices we exploit the triplet motions obtained with Quick triplet algorithm. We performed all the experiments with \( K = 1 \) as with the growing \( K \) we didn’t get much improvement in accuracy but worsened the running times. The \( TL \) was set to 0.15.

Algorithm 1 The Quick triplet algorithm

1: \textbf{Input:} a triplet motion \( \{ R_{21}, R_{23}, R_{31}, R_{32}, R_{33} \} \)
2: \( V_{P1}, V_{P2}, V_{P3} \) vector of points’ correspondences
3: \( N = V_{P1}.\text{size}() = V_{P2}.\text{size}() = V_{P3}.\text{size}() \)
4: \( R_i = [0, 0, 0], R_1 = 1 \)
5: \( C_2 = t_2, R_2 = R_{21} \)
6: \( U_1 = V_{P1}[i] \)
7: \( U_2 = R_2 \cdot V_{P2}[i] \)
8: \( U_3 = V_{P3}[i] \)
9: \( P_{int} = \text{Intersect}(C_1 + U_1, C_2 + U_2) \)
10: \( V_{P.push}(p) \)
11: \textbf{end for}
12: \( C_3 = t_{31} \cdot \text{median}(V_p) \)
13: \( R_3 = \text{NearestRot}(0.5 \cdot (R_{31} + R_{21} \cdot R_{32})) \)

Figure 1. Computation of the perspective center of the "third" image \( C_3 \) in the Quick triplet algorithm. \( C_1 \) and \( C_2 \) are perspective centers of images 1 and 2, respectively. Relative orientations between images 1 and 2 as well as 1 and 3 are known and related by a 7-parameter transformation. (a) A toy example illustrating the intersection of two straight lines defined by vectors \( \hat{P}_0, \hat{P}_1 \) and \( \hat{Q}_0, \hat{Q}_1 \) (cf. On straight lines intersection); (b) intersection of two lines defined by their image observations and respective perspective centers, \( C_1 \) and \( C_2 \); (c) the sought \( C_3 \) lies at the intersection of the vectors \( \mathbf{t}_{31} \) and \( \mathbf{t}_{3P_{int}} \),

Robust triplet algorithm At this stage every edge \( E_{ij} \) in \( \mathcal{H} \) contains a list of at most \( K \) "third" images that were selected in the preceding step. We begin by re-estimating the poses of the "third" images for each \( E_{ij} \) in \( \mathcal{H} \). This estimation is embedded in a RANSAC framework, and uses two direct estimation models: the spatial resection algorithm (Grunert, 1841), and the Tomasi et al. approach to orientating images with long focal lengths (Tomasi and Kanade, 1992). We always test both algorithms and choose the better result by comparing their re-projection errors. We set the number of RANSAC draws to 100, and take a random subset of 500 and 30 corresponding points for spatial resection and long focal length algorithms, respectively. Finally, a per-triplet bundle adjustment is run on the randomly reduced points.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper. https://doi.org/10.5194/isprs-annals-V-2-2020-71-2020 | © Authors 2020. CC BY 4.0 License.
Figure 2. A toy example of structure approximation in 2D and 3D, where (a) shows the initial 2D structure, (b) shows the fitted ellipse, (c) the fictitious 2D structure, and (d) is the equivalent in 3D.

4. STRUCTURE APPROXIMATION

Figure 2 illustrates a toy example of the structure approximation algorithm. Given a set of initial 3D points resulting from per-pair or per-triplet intersection (see On straight lines intersection), we calculate an ellipsoid inscribed in the points. The ellipsoid is represented by its eigenvalues, eigenvalues as well as its center of gravity. These parameters serve to generate a set of new, fictitious 3D points and their corresponding image measurements. In the same vein, Mayer (Mayer, 2014) suggested a point reduction approach to speed-up the merging process in hierarchical SfM.

The ellipsoids are established for both triplets and pairs of images. To begin with, the center of gravity \( \mu_P \) and the covariance matrix \( K_P \) are determined:

\[
\mu_P = \frac{1}{\sum_{i=1}^{N} Pds_{P_i}} \cdot \sum_{i=1}^{N} P_i \cdot Pds_{P_i},
\]

\[
K_P = \frac{1}{\sum_{i=1}^{N} Pds_{P_i}} \cdot \sum_{i=1}^{N} Pds_{P_i}^2 \cdot P_i \cdot P_i^T - \mu_P \cdot \mu_P^T,
\]

where \( P = \{P_i\} \) are the point correspondences, \( Pds \) is the weight function that penalizes large reprojection errors (err) and small base-to-height ratios \( (bb) \), \( Pds = 1/\left[1 + \frac{err^2}{bb} \right] \). In our experiments \( \alpha \) was fixed to 10. Then, the eigenvalues and eigenvectors of the \( K \) matrix are retrieved with the Jacobi method (Press et al., 1992).

We choose to approximate the structure with 5 symmetrically distributed points (see Figure 2(d)) because it is the minimal number of points sufficient for direct pose estimation algorithms such as the 5-point algorithm (essential matrix), spatial resection as well as for a similarity transform that brings two stereo reconstructions to a common coordinate frame (e.g. in hierarchical SFM). Given the eigenvalues \( e \) and eigenvectors \( E_i \), we generate a new point \( H_i \) from:

\[
H_i = \mu + e_1 E_1 \cdot h_{i_1} + e_2 E_2 \cdot h_{i_2} + e_3 E_3 \cdot h_{i_3}
\]

where \( \mu \) is the ellipsoid’s center of gravity; \( E_1, E_2, E_3 \) are first, second and third unit eigenvectors and \( e_1, e_2, e_3 \) are their corresponding eigenvalues; \( \{h_{i_1}, h_{i_2}, h_{i_3}\} \) is the position of a fictitious point in the coordinate frame of the eigenvectors. In our 5-point configuration from Figure 2(d), the following positions were used: \( h_1 = [-1, 1, 1] \), \( h_2 = [1, -1, 1] \), \( h_3 = [1, 1, -1] \), \( h_4 = [0, 0, 0] \), \( h_5 = [-1, -1, -1] \). After the first generation of the new structure, we verify that its eigenvectors and eigenvalues were preserved. If they deviate from their initial values, we recalculate the new structure using an appropriate correction factor. Inspired by Mayer (Mayer, 2014), we additionally extended our experiments to generating 5 randomly distributed points. The reported results are always a mean value over 100 or 1000 repeated random calculations. In the bundle adjustment, the reduced points are weighted according to the number of points that contributed to calculating the ellipsoid using:

\[
Pds = 1 - \frac{N_{bsmax}}{N_{bs}},\ \text{where} \ N_{bs} = \text{the initial points,} \ N_{bsmax} = \text{set to 100.}
\]

5. RESULTS

We run experiments on three groups of datasets: the Strecha benchmark (Strecha et al., 2008), a sequential dataset (CAR) acquired by a mobile mapping system, and a drone dataset (UAV).

To bring the results to the same coordinate frame, we perform a 7-parameter transformation on the cameras’ perspective centers. The Average positional error refers to respective perspective centers’ differences once the transformation has been applied.

Strecha images are provided with their ground truth poses. The CAR dataset contains 647 images that amount to an approximately 2km trajectory without loops (see Figure 3). The camera ground truth poses were calculated using highly accurate ground control points measured in the field with classical surveying techniques (point positional accuracy in the range of few mm). To avoid having to measure many ground control points across the entire trajectory length, the vehicle moved in circles around a block of buildings, and during processing the image correspondences were extracted only between immediate cameras. The SfmInit (Wilson and Snavely, 2014) solution did not succeed in orientating the CAR image set, therefore, we needed to resort to an alternative global SfM available in MicMac (Pierrot Desesilligny and Cléry, 2011). A possible explanation is that

| Dataset      | \( N_{images} \) | \( N_{trips} \) |
|--------------|------------------|----------------|
| F-P11        | 31929            | 1796           |
| HJ-P8        | 17084            | 736            |
| HJ-P25       | 96155            | 8158           |
| C-P10        | 25704            | 1028           |
| C-P19        | 40008            | 2396           |
| C-P30        | 148695           | 7726           |
| CAR          | 2175980         | 157878         |
| UAV          | 170824          | 16388          |

| Image correspondences | \( E_{i_{P+trips}} \) | \( E_{i_{trips}} \) | \( E_{i_{pairs}} \) |
|----------------------|------------------------|----------------------|------------------------|
| F-P11                | 1796                   | 1432                 | 304                   |
| HJ-P8                | 736                    | 540                  | 196                   |
| HJ-P25               | 8158                   | 6638                 | 1529                  |
| C-P10                | 1028                   | 1296                 | 332                   |
| C-P19                | 2396                   | 1850                 | 456                   |
| C-P30                | 7726                   | 6212                 | 1514                  |
| CAR                  | 157878                 | 128688               | -                    |
| UAV                  | 16388                  | 13046                | 3432                  |

Table 3. Number of extracted features in respective datasets.

| Dataset      | \( N_{images} \) | \( N_{trips} \) |
|--------------|------------------|----------------|
| F-P11        | 31929            | 1796           |
| HJ-P8        | 17084            | 736            |
| HJ-P25       | 96155            | 8158           |
| C-P10        | 25704            | 1028           |
| C-P19        | 40008            | 2396           |
| C-P30        | 148695           | 7726           |
| CAR          | 2175980         | 157878         |
| UAV          | 170824          | 16388          |

Table 2. Number of images (\( N_{images} \)) and triplets (\( N_{trips} \)) in respective datasets.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.

https://doi.org/10.5194/isprs-annals-V-2-2020-71-2020 | © Authors 2020. CC BY 4.0 License.
the CAR dataset is a linear acquisition, with potentially many cameras located on the same 3D line. Such acquisitions form a degenerate case for algorithms based only on pairwise constraints. For the same reason the BA refinement on structure approximated with pairs of images (E_p) did not converge. The UAV dataset contains 73 images over an area of 150x160m as illustrated in Figure 5. Analogously to the CAR dataset, the ground truth poses were calculated using ground control points measured in the field. Here as well, SfmInit did not manage to provide an initialisation to the final BA, hence, we resorted to MicMac (Pierrot Deseslligny and Clery, 2011). The followed general pipeline includes:

1. Extraction of correspondences with SIFT (Lowe, 2004).
2. Building of relative motions (Section 4).
3. Approximating the structure (Section 5), variants based on:
   (a) pairs, E_p;
   (b) triplets, E_{tri};
   (c) pairs and triplets, E_{p+tri};
   (d) pairs and triplets with random point distribution, E_{p+tri+rd}.
4. Computation of global motions with SfmInit (Wilson and Snavely, 2014) or MicMac (MM_fusion) (Pierrot Deseslligny and Clery, 2011).
5. Bundle adjustment on camera poses and image correspondences (Pierrot Deseslligny and Clery, 2011), with the same number of iterations;
6. Comparison with the ground truth data (Table 4-5, Figure 3).

Running times Tables 5, 6 and Figure 3 report on the running times. All our experiments were carried out on a machine with Intel Core i7-8550U, 1.8GHz x 8 processor. The Ours approaches concern time spent on the final BA. To deduce the integral processing time, one should add it to the time spent on calculating the initial solution. For a just evaluation, in Table 6 we look at the ratio \(\frac{\text{SfmInitBA}}{\text{Ours}}\), which compares the BA processing times with SIFT to our approximated structure approach. We are faster in all instances by a few factors.

Structure approximation variants Approximating the structure with triplets only (E_{tri}), or triplets and pairs (E_{p+tri}) of images turns out to be the best solution throughout all processed acquisitions. We can assume that for well connected images, which is the case for the three acquisitions tested, using triplets only is accurate enough and fast. For sparser connections, adding pairs may be indispensable. We have also observed a mediocre performance when the structure was approximated only with the pairs (E_p). In general, the refinement step succeeds in improving accuracy with respect to initial poses, and the variants with structure approximated by triplets of images perform at least as good as the sequential SFMs solution.

Deterministic or random fictitious points The results of the three experiments using random points do not stand out from the 5-point configuration. On the Strecha benchmark as well as the UAV dataset, the random distribution is only slightly worse than the best solutions. On the other hand, in the CAR dataset, the random selection sometimes outperforms the other three variants. We perceive that what discriminates the acquisitions is the scene geometry. In the CAR dataset, the scene is characterised by very large depth variations, which allows it to fully model the local structure even of randomly sampled images.

**Limitations** The structure approximation approach is not meant for image sets with unstructured connectivity graphs (e.g. Internet photo collections). The overwhelming redundancy of images does not reduce the information content across the scene, hence, does not accelerate the processing.

### 6. CONCLUSION

There are two leading contributions in this publication. First, we introduce a combinatorial approach to estimating relative motions which enhances the robustness to identified degenerate cases. Second, for a given relative motion, we propose a way of abstracting its 3D structure, such that the motion’s estimation properties are preserved. The abstracted structure injected into a bundle adjustment effectively refines the initial solution at very low computational cost. The concept is validated with respect to ground truth data as well as other software solutions. Future work will concentrate on the purely structureless bundle adjustment, with exclusively view-dependent constraints and a thorough propagation of the minimal structure via relative and absolute motion covariance matrices.

### REFERENCES

Agarwal, S., Snavely, N., Seitz, S. M., Szeliski, R., 2010. Bundle adjustment in the large. *Proc. of ECCV 2010*, Springer, 29–42.

Arie-Nachimson, M., Kovalsky, S. Z., Kemelmacher-Shlizerman, I., Singer, A., Basri, R., 2012. Global motion estimation from point matches. *International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, 81–88.

Cefalu, A., Haala, N., Fritsch, D., 2016. Structureless bundle adjustment with self-calibration using accumulated constraints. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, 3(3).

Chatterjee, A., Govindu, V. M., 2013. Efficient and robust large-scale rotation averaging. *Proc. of CVPR 2013*, 521–528.

Courchay, J., Dalalyan, A., Keriven, R., Sturm, P., 2010. Exploiting loops in the graph of trifocal tensors for calibrating a network of cameras. *Proc. of ECCV 2010*, Springer, 85–99.

Dalalyan, A., Keriven, R., 2009. l_1-Penalized robust estimation for a class of inverse problems arising in multiview geometry. *Advances in Neural Information Processing Systems*, 441–449.

Enqvist, O., Kahl, F., Olsson, C., 2011. Non-sequential structure from motion. *Proc. of ICCV 2011*, 264–271.

Faugeras, O. D., Lustman, F., 1988. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3), 485–508.

Förstner, W., Wrobel, B. P., 2016. *Photogrammetric computer vision*. Springer Nature, Cham.
Table 4. Strecha benchmark. Average position error (err) with ground truth for different sequential SfMs, as well as our approach (Ours). SfM is the initial solution found with [Wilson and Snavely, 2014] that we relied upon in our BA. The $\text{E}_{p+\text{tri}}$ corresponds to the solutions with structure approximated by pairs or triplets with randomly distributed points; a mean value over 1000 repeated calculations is reported; $\text{E}_p$, $\text{E}_{\text{tri}}$, $\text{E}_{p+\text{tri}}$ are the solutions where the structure is approximated with 5 points (Figure 3d), and structure is approximated with pairs, triplets and the fusion of both, respectively.

| Dataset | SfM | Colmap | MM | SfM | Ours | Ours | Ours | Ours |
|---------|-----|--------|----|-----|------|------|------|------|
|         | After BA | $\text{E}_{p+\text{tri}}$ | $\text{E}_{\text{tri}}$ | $\text{E}_{p+\text{tri}}$ | $\text{E}_{p}$ | $\text{E}_{\text{tri}}$ | $\text{E}_{p+\text{tri}}$ | $\text{E}_{p+\text{tri}}$ |
| F-P11   | 1.3 | 1.3    | 33.6 | 2.0 | 2.6 | 2.5 |
| HJ-P8   | 2.7 | 2.8    | 5.7  | 4.4 | 5.9 | 5.5 |
| HJ-P25  | 4.3 | 3.8    | 4.8  | 3.5 | 3.4 |
| C-P10   | 5.5 | 5.7    | 44.4 | 5.9 | 5.5 |
| C-P19   | 20.0| 55.0   | 350.1| 21.3| 56.2|
| C-P30   | 13.3| 16.7   | 263.5| 12.7| 12.6|

Table 5. Strecha benchmark. Running times $\tau$ in conjunction with the results presented in Table 4. Note that the times for Ours concern only the BA step. Times markes with (*) correspond to the total incremental SfM.

| Dataset | SfM | Colmap | MM | SfM | Ours | Ours | Ours | Ours |
|---------|-----|--------|----|-----|------|------|------|------|
|         | $\text{E}_{p+\text{tri}}$ | $\text{E}_{\text{tri}}$ | $\text{E}_{p+\text{tri}}$ | $\text{E}_p$ | $\text{E}_{\text{tri}}$ | $\text{E}_{p+\text{tri}}$ | $\text{E}_{p+\text{tri}}$ |
| F-P11   | 1.3 | 1.3    | 33.6 | 0.7 | 0.6  | 0.7 |
| HJ-P8   | 1.7 | 0.4    | 0.3  | 0.3 | 0.4  | 0.4 |
| HJ-P25  | 8.9 | 2.5    | 1.5  | 2.1 |
| C-P10   | 1.7 | 0.9    | 0.6  | 0.6 |
| C-P19   | 8.3 | 1.7    | 0.8  | 1.5 |
| C-P30   | 9.7 | 4.5    | 1.4  | 3.8 |

Table 6. UAV dataset. Average position error (err) and running times ($\tau$) of MM and the MM is the global and sequential SfMs in MicMac. MM in BA corresponds to the BA refinement on the initial solution. Note that the times for Ours concern only the BA step. In $\text{E}_{p+\text{tri}}$ a mean value over 1000 repeated calculations is reported.

| Dataset | SfM | Colmap | MM | SfM | Ours | Ours | Ours | Ours |
|---------|-----|--------|----|-----|------|------|------|------|
|         | $\text{E}_{p+\text{tri}}$ | $\text{E}_{\text{tri}}$ | $\text{E}_{p+\text{tri}}$ | $\text{E}_p$ | $\text{E}_{\text{tri}}$ | $\text{E}_{p+\text{tri}}$ | $\text{E}_{p+\text{tri}}$ |
| F-P11   | 21.0| 9.8    | 6.2  | 5.3 |
| HJ-P8   | 27.4| 3.8    | 5.3  |
| HJ-P25  | 18.2| 5.3    |
| C-P10   | 17.0| 7.2    |
| C-P19   | 17.0| 7.2    | 5.3  |
| C-P30   | 17.0| 7.2    |

Indelman, V., Roberts, R., Beall, C., Dellaert, F., 2012. Incremental light bundle adjustment. Georgia Institute of Technology.

Jeong, Y., Nister, D., Steedly, D., Szeliski, R., Kweon, I. S., 2011. Pushing the envelope of modern methods for bundle adjustment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8), 1605–1617.

Jiang, N., Cui, Z., Tan, P., 2013. A global linear method for camera pose registration. *Proc. of CVPR 2013*, 481–488.

Julia, F., N. Monasse, P., 2017. A critical review of the trifocal tensor estimation. *Pacific-Rim Symposium on Image and Video Technology*, 337–349.

Kahl, F., Hartley, R., 2008. Multiple-View Geometry Under the L-Infinity-Norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9), 1603–1617.

Klopschitz, M., Irshara, A., Reitmayr, G., Schmalstieg, D., 2010. Robust incremental structure from motion. *Proc. 3DPVT*, 1–8.

Kushal, A., Agarwal, S., 2012. Visibility based preconditioning for bundle adjustment. *Proc. of CVPR 2012*, 1442–1449.

Kühn, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110.

Martinec, D., Pajdla, T., 2007. Robust rotation and translation estimation in multiview reconstruction. *Proc. of CVPR 2007*, 1–8.

Mayer, H., 2014. Efficient hierarchical triplet merging for camera pose estimation. *German Conference on Pattern Recognition*, Springer, 399–409.

Govindu, V. M., 2001. Combining two-view constraints for motion estimation. *Proc. of CVPR 2011*.

Govindu, V. M., 2006. Robustness in motion averaging. *Proc. of ACCV* 2006, 457–466.

Grunert, J., 1841. Das pothenotische problem, in erweiterter gestalt, nebst bemerkungen über seine anwendung in der Geodäsie. *Archiv der Mathematik und Physik*, 238–248.

Hartley, R., Aftab, K., Trumpf, J., 2011. L1 rotation averaging using the weiszfeld algorithm. *Proc. of CVPR 2011*, 3041–3048.

Hartley, R., Trumpf, J., Dai, Y., Li, H., 2013. Rotation averaging. *International Journal of Computer Vision*, 103(3), 267–305.

Hartley, R., Zisserman, A., 2003. *Multiple view geometry in computer vision*. Cambridge university press.
Moulon, P., Monasse, P., Marlet, R., 2013. Global fusion of relative motions for robust, accurate and scalable structure from motion. Proc. of ICCV 2013, 3248–3255.

Moulon, P., Monasse, P., Perrot, R., Marlet, P., 2016. OpenMVG: Open multiple view geometry. International Workshop on Reproducible Research in Pattern Recognition, Springer, 60–74.

Olsson, C., Eriksson, A., Hartley, R., 2010. Outlier removal using duality. Proc. of CVPR 2010, 1450–1457.

Pierrot Deseilligny, M., Cléry, I., 2011. Apero, an open source bundle adjustment software for automatic calibration and orientation of set of images. Proceedings of the ISPRS Symposium, 3DARCH11, 269277.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., 1992. Numerical recipes in C++. The art of scientific computing, 2, 1002.

Rodriguez, A. L., López-de Teruel, P. E., Ruiz, A., 2011. GEA optimization for live structureless motion estimation. Proc. of ICCV Workshop 2011, 715–718.

Rupnik, E., Daakir, M., Pierrot Deseilligny, M., 2017. MicMac—a free, open-source solution for photogrammetry. Open Geospatial Data, Software and Standards, 2(1), 14.

Schonberger, J. L., Frahm, J., 2016. Structure-from-motion revisited. Proc. of CVPR 106, 4104–4113.

Sim, K., Hartley, R., 2006. Removing outliers using the l∞ norm. Proc. of CVPR 2006, 1, 485–494.

Snavely, N., Seitz, S. M., Szeliski, R., 2006. Photo tourism: exploring photo collections in 3D. ACM Transactions on Graphics, 25, 835–846.

Steffen, R., Frahm, J.-M., Förstner, W., 2010. Relative bundle adjustment based on trifocal constraints. Proc. of ECCV 2010, 282–295.

Strecha, C., Von Hansen, W., Van Gool, L., Fua, P., Thoenissen, U., 2008. On benchmarking camera calibration and multi-view stereo for high resolution imagery. Proc. of CVPR 2008, 1–8.

Sweeney, C., Sattler, T., Hollerer, T., Turk, M., Pollefeys, M., 2015. Optimizing the viewing graph for structure-from-motion. Proc. of ICCV 2015, 801–809.

Toldo, R., Gherardi, R., Farenzena, M., Fusiello, A., 2015. Hierarchical structure-and-motion recovery from uncalibrated images. Computer Vision and Image Understanding, 140, 127–143.

Tomasi, C., Kanade, T., 1992. Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision, 9(2), 137–154.

Triggs, B., McLauchlan, P. F., Hartley, R. I., Fitzgibbon, A. W., 1999. Bundle adjustment—a modern synthesis. International workshop on vision algorithms, Springer, 298–372.

Wilson, K., Snavely, N., 2014. Robust global translations with 1DSFM. Proc. of ECCV 2014, 61–75.
Figure 3. CAR dataset. Average position error ($\text{err}$) along the trajectory w.r.t. ground truth for several structure approximation variants: $E_{p+tri}$ in black dashed line, $E_{p+tri}$ in continuous magenta line, $E_{tri}$ in dotted blue line. MM (green dashed line) and MM$_{Init}$ (green dash-dotted line) are the sequential and global SfMs in MicMac (Pierrot Desessigny and Clercy, 2011), and in red continuous like it is Colmap’s SfM (Schonberger and Frahm, 2016). MicMac and Colmap results were computed with SIFT (Lowe, 2004). In variant $E_{p+tri}$ a mean value over 100 repeated calculations is reported. The high frequency drift is due to the cyclic nature of the acquisition geometry.

Figure 4. Left: excerpt from the CAR dataset. Right: (a) and (c) is the ground truth in top and side view; (b) and (d) is the equivalent $E_{p+tri}$ result. Note the multiple facade effect in (b) and (d) due to error accumulation, and leads to a low frequency drift, as plotted in Figure 3.

Figure 5. Left: excerpt from the UAV dataset. Right: (a) and (c) is the ground truth in oblique and side view; (b) and (d) is the equivalent SfmInit result. Note the erroneous initialisation of SfmInit in (b) and (d) where exclusively pairwise constraints are used.