Robust and Efficient Camera Motion Synchronization via Matrix Decomposition

Federica Arrigoni[1](✉), Beatrice Rossi[2], and Andrea Fusiello[1]

[1] DIEGM, Università di Udine, Via Delle Scienze, 208, Udine, Italy
arrigoni.federica@spes.uniud.it
[2] AST Lab, STMicroelectronics, Via Olivetti, 2, Agrate Brianza, Italy

Abstract. In this paper we present a structure-from-motion pipeline based on the synchronization of relative motions derived from epipolar geometries. We combine a robust rotation synchronization technique with a fast translation synchronization method from the state of the art. Both reduce to computing matrix decompositions: low-rank & sparse and spectral decomposition. These two steps successfully solve the motion synchronization problem in a way that is both *efficient* and *robust* to outliers. The pipeline is global for it considers all the images at the same time. Experimental validation demonstrates that our pipeline compares favourably with some recently proposed methods.

Keywords: Structure from motion · Synchronization · Low-rank decomposition

1 Introduction

Structure from Motion (SfM) is a crucial problem in Computer Vision. The goal is to recover both 3D structure, namely 3D coordinates of scene points, and motion parameters, namely attitude (rotation) and position of the cameras, starting from image point correspondences.

For many years, most practical SfM pipelines have adopted either *sequential* or *hierarchical* approaches. Sequential methods, such as [20], incrementally increase a partial reconstruction by iteratively adding new cameras and 3D points, whereas hierarchical ones, such as [22], organize images in a binary tree and progressively merge smaller reconstructions into larger ones. Although being highly accurate, these approaches suffer from two main disadvantages: on one hand they require computationally-expensive intermediate bundle adjustment minimizations to contain error propagation, on the other hand the final reconstruction may depend on the order in which cameras are added or on the choice of the initial pair.

Recently, *global* SfM pipelines, such as [1,15,17–19], have gained increasing attention in the community. Such methods start from the relative motions, i.e. epipolar geometries computed from point matches among the images, compute the angular attitude and position of the cameras with respect to an absolute coordinate frame, and then recover the 3D structure. Here the term global means

V. Murino and E. Puppo (Eds.): ICIAP 2015, Part I, LNCS 9279, pp. 444–455, 2015.
DOI: 10.1007/978-3-319-23231-7_40

that such techniques take into account the entire relative motion information at once, or, in other terms, they consider the whole *epipolar graph*, which has a vertex for each camera and edges in correspondence of view pairs having consistent matching points. Global methods have the advantage of fairly distributing errors among the cameras, and thus they need bundle adjustment refinement only at the end, thereby performing faster than the other methods.

The core of global methods is the so-called *motion synchronization problem* (a.k.a *motion registration* or *motion averaging*), i.e. computing *absolute* positions and attitudes starting from *relative* measurements. Formally, the goal is to compute n rotation matrices $R_i \in SO(3)$ and n translation vectors $\mathbf{t}_i \in \mathbb{R}^3$ such that the projection matrix of the i-th camera is expressed as $P_i = K_i \left[R_i \, \mathbf{t}_i\right]$, where $K_i \in \mathbb{R}^{3\times3}$ are the internal calibration matrices, assumed known. It is inherent to the problem that the motion parameters can be recovered up to a roto-translation and a single scaling factor. Most techniques split such a problem in two stages: first they compute the absolute attitude of each camera, and then they recover camera positions.

The first stage is known as *rotation synchronization* (or *rotation registration* or *multiple rotation averaging*) and a thorough overview of the theory behind it can be found in [12]. Several approaches have been proposed to solve this problem, both within SfM pipelines and in stand-alone works. Non-robust methods, such as [1,17,18], suffer from the presence of inconsistent/outlier pairwise information, i.e. skewed epipolar geometries caused by mismatches, and thus they need a computationally-expensive preliminary step devoted to detect and remove such outliers. On the contrary, robust techniques, such as [2,8,11], are inherently resilient to outliers and hence they are more efficient.

The position recovery stage (a.k.a. *translation synchronization* or *translation registration*) can use only constraints derived from relative translation directions, such as [4,9,17,19], or additionally exploit point correspondences among the images, such as [1,15,24]. In practical SfM pipelines, methods from the former category should be preferred: besides being more consistent with the structure *from* motion paradigm – where structure comes into play only *after* motion has been computed – they are potentially more efficient, since they reduce memory usage.

Contribution. In this paper we combine the rotation synchronization technique in [2] with the translation synchronization method in [4]. The resulting global pipeline successfully solves the motion synchronization problem, while ensuring at the same time both *efficiency* and *robustness* to outliers. More precisely, motion synchronization is reduced to computing two matrix decompositions, involving matrices of dimension $3n \times 3n$: first a low-rank & sparse decomposition, then a spectral factorization. Experimental validation demonstrates that our pipeline compares favourably with some recently proposed methods.

1.1 Overview

The proposed SfM pipeline is organized as follows.

Step 1: Computing Relative Motions. First, a collection of reliable correspondences for each image pair is obtained by extracting and matching SIFT features. After expressing these image points in normalized coordinates (i.e. left-multiplying by the inverse of the calibration matrices), the essential matrices are computed through RANSAC in combination with the 8-point algorithm. The *epipolar graph* is then built with an edge linking two views for which a sufficient number of inliers have been found. For each edge the relative motion is computed from the essential matrix, and it is subsequently refined through Bundle Adjustment (BA). The X84 rejection rule [10] is introduced at each step of BA, removing image points with the highest reprojection error.

Step 2: Motion Synchronization. The first step synchronizes relative rotations in a robust manner, and it is at the same time efficient, thanks to the usage of a faster alternative to singular value decomposition for computing low-rank projections (Section 2). The second step (Section 3) reduces translation recovery to a graph embedding problem, which is equivalent to computing the smallest eigenvector of a data matrix, which does not involve corresponding points, resulting in an extremely fast method. The relative translation directions are refined through Iteratively Reweighted Least Squares (IRLS).

Step 3: Final Refinement. The correspondences are tracked through the images and 3D coordinates of scene points are computed by triangulation. The structure and absolute translations are refined with a partial BA with fixed rotations. Then, a global BA is applied to improve the quality of structure and motion estimation. The idea of using a two-stage BA is inspired by [17,18] and it is motivated by the fact that rotations are more reliable in general. As in Step 1, at each iteration of BA the X84 rejection rule singles out outliers, based on the reprojection error.

2 Rotation Synchronization

The rotation synchronization step in a global structure-from-motion pipeline takes as input the observed pairwise rotations $\widehat{R}_{ij} \in SO(3)$ and returns the absolute rotations of the cameras $R_i \in SO(3)$ such that the latter are "compatible" with the former, i.e. $R_i R_j^\mathsf{T} \approx \widehat{R}_{ij}$. In this paper we use the hat accent to denote noisy measurements. The notion of compatibility can be formalized by considering the chordal distances between the estimated and unknown relative rotations, resulting in the following rotation synchronization problem

$$\min_{R_i \in SO(3)} \sum_{(i,j)\in\mathcal{E}} \left\| \widehat{R}_{ij} - R_i R_j^\mathsf{T} \right\|_F^2 . \tag{1}$$

where $\mathcal{E} \subseteq \{1,\ldots,n\} \times \{1,\ldots,n\}$ is the edge set of the epipolar graph.

More precisely, we use the R-GoDec Algorithm introduced in [2] which solves a regularized version of (1) in order to cope with outlying relative rotations. In Section 2.1 we describe such an algorithm in a general scenario, while in Section 2.2 we explain how to apply this method to find camera absolute rotations.

2.1 The R-GoDec Algorithm

The *matrix completion* problem [5,7,14] consists in completing a low-rank matrix $\widehat{X}$ starting from an incomplete subset of its entries $\mathcal{P}_{\Omega}(\widehat{X})$ possibly corrupted with a low level of noise. Here Ω denotes the sampling set and $\mathcal{P}_{\Omega}$ is the projection onto the space of matrices that vanish outside Ω. The goal of *low-rank and sparse matrix decomposition* [6,25] is to find a low-rank term L, a sparse term S representing outlier measurements, and a noise term N such that a data matrix $\widehat{X}$ can be written as

$$\widehat{X} = L + S + N. \tag{2}$$

On one hand, matrix completion techniques are able to guess missing entries, but they are not robust to outliers. On the other hand, matrix decomposition techniques handle sparse errors of large intensity but they do not deal with missing data. There is a small fraction of methods (including [2,13,23]) addressing this double problem simultaneously, i.e. performing robust matrix completion or equivalently matrix decomposition with missing entries.

The R-GoDec Algorithm [2] is a combination of matrix completion and matrix decomposition techniques, and was derived by properly modifying the GoDec Algorithm [25] in order to handle outliers and missing entries simultaneously. More precisely, the sparse term S in (2) is replaced by the sum of two terms S_1 and S_2 having dual supports: S_1 is a sparse matrix over the sampling set Ω which is nonzero in correspondence of the outlier entries only; S_2 has support on Ω^{C} (the complementary of Ω) and it is an approximation of $-\mathcal{P}_{\Omega^{\mathsf{C}}}(L)$, representing recovery of missing entries. This results in the following model

$$\widehat{X} = L + S_1 + S_2 + N. \tag{3}$$

Assuming that the rank r of the low-rank term is known in advance, the associated minimization problem is

$$\begin{gathered}\min_{L,S_1,S_2} \frac{1}{2} \left\| \widehat{X} - L - S_1 - S_2 \right\|_F^2 + \lambda \left\| S_1 \right\|_1 \\ \text{s.t. } \operatorname{rank}(L) \le r, \quad \operatorname{supp}(S_1) \subseteq \Omega, \quad \operatorname{supp}(S_2) = \Omega^{\mathsf{C}}\end{gathered} \tag{4}$$

where $\lambda \geq 0$ is a regularization parameter, and $\|S_1\|_1$ denotes the ℓ^1-norm of its argument considered as a vector. Since the ℓ^1-norm is a sparsity-inducing norm, it is expected to separate sparse outliers from non corrupted low-rank data by minimizing the cost function in (4).

In order to solve problem (4), R-GoDec alternatively minimizes the cost function with respect to each optimization variable, keeping constant the others. In other words, the following steps are iterated until convergence.

- The rank-r approximation of $\widehat{X} - S_1 - S_2$ is assigned to L;
- The minimizer of the cost function in (4) with respect to S_1 is assigned to S_1, i.e. the result of applying entry-wise *Soft Thresholding* [3] with parameter λ to the matrix $\mathcal{P}_{\Omega}(\widehat{X} - L)$;
- The quantity $\mathcal{P}_{\Omega^{\mathsf{c}}}(\widehat{X} - L - S_1) = -\mathcal{P}_{\Omega^{\mathsf{c}}}(L)$ is assigned to S_2.

The low-rank projection is computed through *Bilateral Random Projections* (BRP) [25] instead of Singular Value Decomposition (SVD) in order to reduce the computational cost. More details can be found in [2].

2.2 Robust Rotation Synchronization

Let us introduce the following notation:

$$R = \begin{bmatrix} R_1 \\ R_2 \\ \dots \\ R_n \end{bmatrix} \in \mathbb{R}^{3n \times 3}, \quad X = \begin{pmatrix} I & R_{12} & \dots & R_{1n} \\ R_{21} & I & \dots & R_{2n} \\ \dots & & & \dots \\ R_{n1} & R_{n2} & \dots & I \end{pmatrix} \in \mathbb{R}^{3n \times 3n}. \tag{5}$$

As observed in [1], it follows from the compatibility constraint $R_{ij} = R_i R_j^{\mathsf{T}}$ that the block matrix X admits the factorization $X = RR^{\mathsf{T}}$ and hence it has rank 3. Let $\widehat{X}$ be an estimate of X, constructed by replacing R_{ij} with $\widehat{R}_{ij}$ in (5). Matrix completion is required here since not all $\widehat{R}_{ij}$ are available in practice, i.e. the epipolar graph is not complete. Moreover, matrix decomposition is required since some pairwise rotations may be wrong due to repetitive patterns and symmetries in the images. Indeed, these structures generate false essential matrices, namely two-view geometries which do not agree with the real 3D geometry, even if they are satisfied by the majority of point matches. Thus, in order to handle both missing and outlier blocks in $\widehat{X}$, in addition to a diffused noise, a decomposition of the form (3) is required, and it can be computed through the R-GoDec Algorithm with $r = 3$.

Formally, computing the low-rank & sparse matrix decomposition of $\widehat{X}$ is equivalent (up to a relaxation) to solve the rotation synchronization problem (1) in a robust manner. We now briefly explain this connection, more details can be found in [2]. By using the notation in (5), it is straightforward to see that problem (1) can be expressed equivalently as

$$\begin{gathered} \min_{X} \frac{1}{2} \left\| \mathcal{P}_{\Omega}(\widehat{X} - X) \right\|_F^2 \\ \text{s.t. } X = RR^{\mathsf{T}},\ R \in SO(3)^n \end{gathered} \tag{6}$$

which, if all the requirements on X are ignored but the rank constraint, reduces to

$$\begin{gathered} \min_{L} \frac{1}{2} \left\| \mathcal{P}_{\Omega}(\widehat{X} - L) \right\|_F^2 \\ \text{s.t. } \operatorname{rank}(L) \leq 3. \end{gathered} \tag{7}$$

The notation L instead of X highlights that L will not coincide with X in general, due to the rank relaxation, i.e. L will not be symmetric positive semidefinite and composed of rotations. Problem (7) is a matrix completion problem [7], and it can be written in an equivalent form as follows

$$\begin{aligned} &\min_{L,S_2} \frac{1}{2} \left\| \widehat{X} - L - S_2 \right\|_F^2 \\ &\text{s.t. } \operatorname{rank}(L) \leq 3, \quad \operatorname{supp}(S_2) = \Omega^C \end{aligned} \tag{8}$$

where the additional variable S_2 is introduced to handle missing entries and the projection operator $\mathcal{P}_\Omega$ is not required. Finally, if robustness is introduced in (8) through ℓ^1-regularization, then problem (4) is obtained.

Once problem (4) is solved by means of the R-GoDec Algorithm, the optimal L is used to estimate the absolute rotations. Since the solution of rotation synchronization is defined up to a global rotation, any block-column of L – after projection onto $SO(3)$ – can be viewed as an estimate of R. The absolute rotations computed in this way are resilient to outliers, since the cost function in (4) naturally includes the outliers in its definition through the sparse matrix S_1. However, *a posteriori* outlier detection is useful for the subsequent step of translation synchronization. Rogue relative rotations correspond to non-zero entries in the sparse matrix S_1. Thus a rotation $\widehat{R}_{ij}$ is classified as outlier if the number of non-zero entries in the associated 3×3 block in S_1 is greater than a given threshold θ, with $\theta \in \{1, \ldots, 9\}$. In this case, the edge (i, j) is removed from $\mathcal{E}$, since the entire epipolar geometry associated to (i, j) is likely to be wrong.

3 Translation Synchronization

The translation synchronization step in a global structure-from-motion pipeline takes as input either a set of corresponding points or the relative translation directions, and returns the absolute translations of the cameras $\mathbf{t}_i \in \mathbb{R}^3$, or equivalently the camera positions (centers) $\mathbf{c}_i = -R_i^{\mathsf{T}} \mathbf{t}_i$. Accordingly, there are several ways to define a suitable cost function for the problem. A possibility is to constraint camera locations to be linear combinations of rays emanating from their neighbours, with known directions and unknown coefficients. This concept is formalized in [4] where a fast spectral solution is developed. In Section 3.2 we describe such algorithm, while in Section 3.1 we explain how to refine the translation directions in an accurate way, based on the knowledge of absolute rotations and corresponding points.

3.1 Refining the Relative Translation Directions

First, the relative rotations are updated by using the compatibility constraint $R_{ij} = R_i R_j^{\mathsf{T}}$, where R_i are the absolute rotations returned by R-GoDec. Then, the epipolar constraint with *known* rotation becomes a linear equation in the unknown $\mathbf{t}_{ij} \in \mathbb{R}^3$ for each pair of point matches.

Let $\{\mathbf{p}_i^k, \mathbf{p}_j^k\}_{k=1}^{N_{ij}}$ denote a set of N_{ij} corresponding points for the pair $(i,j) \in \mathcal{E}$ expressed in normalized coordinates. By using the invariance to permutation of a triple product (up to sign), the epipolar constraint of this image pair can be expressed equivalently as

$$(\mathbf{p}_i^k \times R_{ij}\mathbf{p}_j^k)^\mathsf{T}\mathbf{t}_{ij} = 0. \tag{9}$$

By stacking all these equations, a homogeneous system is obtained, whose solution is the desired estimate of the relative translation direction $\mathbf{t}_{ij}/\left\|\mathbf{t}_{ij}\right\|$. In order to cope with rogue point correspondences we apply *Iteratively Reweighted Least Squares* (IRLS) to the residuals e_k of the linear system (9). The weights w_k are computed by using Mosteller and Tukey's weight function [16], namely $w_k = (1-(e_k/s)^2)^2$ if $e_k \leq s$, $w_k = 0$ otherwise.

3.2 A Fast Spectral Method

Let $\mathbf{c}_{ij} = \mathbf{c}_i - \mathbf{c}_j = -R_i^\mathsf{T}\mathbf{t}_{ij}$ denote the baseline of the pair (i,j) and let $\mathbf{d}_{ij} = \mathbf{c}_{ij}/\left\|\mathbf{c}_{ij}\right\| = -R_i^\mathsf{T}\mathbf{t}_{ij}/\left\|\mathbf{t}_{ij}\right\|$ denote its direction. Let $\widehat{\mathbf{d}}_{ij}$ be an estimate of $\mathbf{d}_{ij}$, computed as explained in the previous section. The goal is to find a realization of the locations $\mathbf{c}_i \in \mathbb{R}^3$ starting from the measurements $\widehat{\mathbf{d}}_{ij}$. In [4] camera positions are recovered by imposing that camera-to-camera displacements $(\mathbf{c}_i - \mathbf{c}_j)$ are maximally "consistent" with the constraint directions $\widehat{\mathbf{d}}_{ij}$. The notion of consistency is expressed as a minimum-squared-error problem where the components of the displacements that are orthogonal to the constraints are minimized. This results in the following problem

$$\min_{\mathbf{c}_i \in \mathbb{R}^3} \sum_{(i,j)\in\mathcal{E}} \left\| (\mathbf{c}_i - \mathbf{c}_j)^\mathsf{T} \widehat{K}_{ij} \right\|_F^2 \tag{10}$$

where $\widehat{K}_{ij}$ is an orthonormal basis for the kernel of $\widehat{\mathbf{d}}_{ij}$. Optionally, weights can be included in (10) to reflect the uncertainty of the estimates $\widehat{\mathbf{d}}_{ij}$ (see [4] for details).

If $\mathbf{c} = [\mathbf{c}_1^\mathsf{T}, \ldots, \mathbf{c}_n^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{3n}$ denotes the stack of the unknown locations $\mathbf{c}_i$, then the following equalities hold for the cost function in (10)

$$\begin{gathered} \sum_{(i,j)\in\mathcal{E}} \left\| (\mathbf{c}_i - \mathbf{c}_j)^\mathsf{T} \widehat{K}_{ij} \right\|_F^2 = \sum_{(i,j)\in\mathcal{E}} (\mathbf{c}_i - \mathbf{c}_j)^\mathsf{T} \widehat{K}_{ij}\widehat{K}_{ij}^\mathsf{T} (\mathbf{c}_i - \mathbf{c}_j) = \\ \sum_{(i,j)\in\mathcal{E}} \mathbf{c}_i^\mathsf{T}\widehat{D}_{ij}\mathbf{c}_i + \mathbf{c}_j^\mathsf{T}\widehat{D}_{ij}\mathbf{c}_j - \mathbf{c}_i^\mathsf{T}\widehat{D}_{ij}\mathbf{c}_j - \mathbf{c}_j^\mathsf{T}\widehat{D}_{ij}\mathbf{c}_i = 2\mathbf{c}^\mathsf{T}\widehat{H}\mathbf{c} \end{gathered} \tag{11}$$

where $\widehat{D}_{ij} = I_3 - \widehat{\mathbf{d}}_{ij}\widehat{\mathbf{d}}_{ij}^\mathsf{T} = \widehat{K}_{ij}\widehat{K}_{ij}^\mathsf{T} \in \mathbb{R}^{3\times 3}$ is the orthogonal projector onto $\operatorname{Ker}(\widehat{\mathbf{d}}_{ij})$, $\widehat{D} \in \mathbb{R}^{3n\times 3n}$ is constructed by placing $\widehat{D}_{ij}$ in each 3×3 block (and zero blocks in correspondence of missing edges), and $\widehat{H} = \operatorname{blockdiag}(\widehat{D}(\mathbf{1}_n \otimes I_3)) - \widehat{D}$. Here $\mathbf{1}_n$ denotes the vector in $\mathbb{R}^n$ with 1 at each entry, and $\otimes$ denotes the

Kronecker product. Thus problem (10) is equivalent to minimize the following quadratic form

$$\min_{\|\mathbf{c}\|=1} \mathbf{c}^{\mathsf{T}} \widehat{H} \mathbf{c}. \tag{12}$$

Problem (12) admits a closed-form solution which is the eigenvector of $\widehat{H}$ with minimum eigenvalue. However, $\mathbf{c}_i = \mathbf{c}_j$ for all i, j is a trivial solution to problem (10), that corresponds to mapping all the cameras to a single point in $\mathbb{R}^3$. This trivial subspace is spanned in $\mathbb{R}^{3n}$ by the vectors $\left[1\ 0\ 0 \ldots 1\ 0\ 0\right]^{\mathsf{T}}$, $\left[0\ 1\ 0 \ldots 0\ 1\ 0\right]^{\mathsf{T}}$, $\left[0\ 0\ 1 \ldots 0\ 0\ 1\right]^{\mathsf{T}}$ which can be concatenated to form the matrix $\mathbf{1}_n \otimes I_3 \in \mathbb{R}^{3n \times 3}$. Thus the kernel of $\widehat{H}$ will have (exactly or approximately) dimension 4, and the sought solution must belong to $\text{Ker}(\widehat{H})$ and be orthogonal to $\mathbf{1}_n \otimes I_3$ at the same time, in order to avoid the trivial solution. To compute it, it is sufficient to project $\widehat{H}$ onto an orthogonal basis $Q \in \mathbb{R}^{3n \times 3n-3}$ of $\text{Ker}(\mathbf{1}_n \otimes I_3)$, compute the eigen-decomposition of the reduced problem and then back project the eigenvectors.

This method has the advantage of being both simple and extremely fast, as translation synchronization is cast to an eigenvalue decomposition of a matrix whose size does not depend on the number of matching points. More details about this technique can be found in [4], including problem pathologies that appear where the data are insufficient or inconsistent.

4 Experiments

In this section we evaluate our pipeline on publicly available datasets [21] where the number of cameras varies from 8 to 30 and ground-truth motions are available. All the experiments are carried out in MATLAB on a dual-core 1.3 GHz machine.

To define the epipolar graph, we consider only image pairs having more than 500 inlier correspondences. As for rotation averaging, we perform at most 100 iterations of R-GoDec, using the value $\lambda = 0.05$, and we choose the value $\theta = 3$ for outlier detection. In order to compare our results with ground-truth absolute rotations, we find the optimal rotation that aligns them by performing *single* rotation averaging [11]. As for camera positions, we find the scale and translation of the optimal alignment by solving the associated linear system in the least-square sense. We use the angular distance and the euclidean norm as distance measures for rotations and positions respectively. The results of our simulations are reported in Tables 1, 2 and 3, where our pipeline is compared with the global methods described in [17,19]. As for [19], the online code concerns motion averaging only, thus we used our pipeline for the remaining steps. The results of Moulon et al. reported in Table 2 are taken from their original paper [17], where only translation errors are disclosed. We also include in the comparison the hierarchical approach of [22], whose binary code is available online. To evaluate the execution times, we consider the largest datasets, i.e. HerzJesu-P25 and Castle-P30, and MATLAB implementations.

Table 1. Mean angular errors [degrees] on the absolute rotations.

	Our Pipeline		Ozyesil et al. [19]		Samantha [22]
	before BA	after BA	before BA	after BA	
Castle-P30	0.78	0.05	1.97	0.05	0.06
Castle-P19	1.57	0.05	3.69	0.05	0.09
Entry-P10	0.44	0.03	0.56	0.04	0.05
Fountain-P11	0.03	0.03	0.03	0.03	0.06
HerzJesu-P25	0.13	0.04	0.14	0.04	0.03
HerzJesu-P8	0.04	0.03	0.06	0.03	0.04

Table 2. Mean errors [meters] on the absolute positions.

	Our Pipeline		Ozyesil et al. [19]		Moulon et al. [17]	Samantha [22]
	before BA	after BA	before BA	after BA	after BA	
Castle-P30	1.123	0.030	1.393	0.030	0.022	0.033
Castle-P19	1.493	0.036	1.769	0.032	0.026	0.046
Entry-P10	0.433	0.009	0.203	0.010	0.006	0.022
Fountain-P11	0.006	0.003	0.004	0.003	0.003	0.006
HerzJesu-P25	0.038	0.009	0.065	0.009	0.005	0.031
HerzJesu-P8	0.009	0.004	0.007	0.005	0.004	0.007

Table 3. Execution times [seconds] of motion synchronization.

	Our pipeline		Ozyesil et al. [19]	
	Rotation	Translation	Rotation	Translation
Castle-P30	0.05	0.05	0.15	0.80
HerzJesu-P25	0.04	0.04	0.13	1.32

Tables 1 and 2 show that our pipeline is able to recover camera motion accurately, achieving results which are comparable to the other analysed techniques, and within the accuracy of the ground truth [21]. We obtain an average angular error less than 0.1 degrees and an average location error of the order of millimeters, after the final Bundle Adjustment (BA), confirming that motion synchronization provides a good initialization. In some cases the result is more than an initialization, being already very close to the BA optimum. In some other cases (namely, Castle-P*), the difference with BA is higher. Nevertheless, the angular errors obtained with our pipeline before BA are lower than those obtained with [19], confirming the effectiveness of low-rank & sparse decomposition for outlier handling.

As concerns the execution cost, our method outperforms the technique in [19]. Indeed, the method used in our pipeline, is one of the fastest translation synchronization techniques present in the literature as it finds camera positions by eigenvalue-decomposition of a $3n \times 3n$ matrix. Also the rotation synchronization is very efficient, as the R-GoDec Algorithm is based on fast BRP. We cannot directly compare the performances of [17], as the code is in C++, but we draw the attention of the reader on the outlier removal step, which consists in performing Bayesian inference on cycles within the epipolar graph, analysing the deviation from the identity. The number of cycles analysed must be high in order to make meaningful statistical inference, resulting in a computationally expensive technique.

Finally, in Figure 1 we report the 3D point cloud obtained with our system in the case of the Castle-P30 sequence. Even if these images contain repetitive windows, resulting in outlying two-view geometries, we are able to recover the 3D structure accurately.

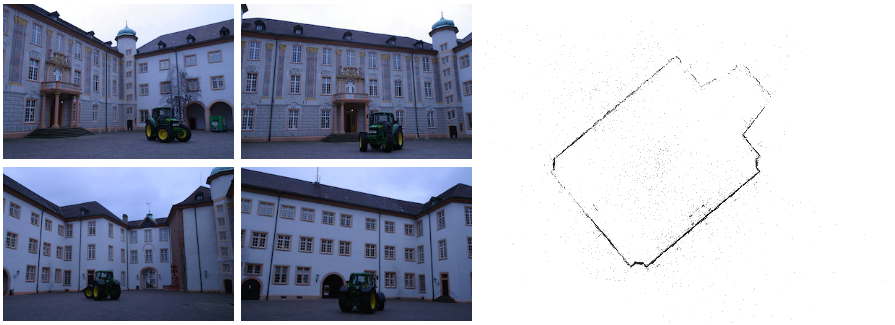

Fig. 1. Left: sample images of the Castle-P30 dataset. Right: sparse 3D reconstruction obtained with our pipeline. The root-mean-squared reprojection error (RMSE) is 0.1681 pixels.

5 Conclusion

In this paper we proposed a global SfM pipeline, based on the synchronization of relative motions. We combined a robust rotation synchronization technique with a fast translation synchronization method from the state of the art. Absolute rotations are computed through low-rank & sparse matrix decomposition (R-GoDec), while absolute locations are recovered through eigenvalue decomposition. The resulting system inherits robustness from R-GoDec and efficiency from both matrix decompositions. Thus it is able to recover camera motion accurately, even in the presence of outliers, achieving low computational cost, as demonstrated by the experiments.

References

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

2. Arrigoni, F., Rossi, B., Magri, L., Fragneto, P., Fusiello, A.: Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In: International Conference on 3D Vision, pp. 491–498 (2014)
3. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences **2**(1), 183–202 (2009)
4. Brand, M., Antone, M., Teller, S.: Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In: Pajdla, T., Matas, J.G. (eds.) ECCV 2004. LNCS, vol. 3022, pp. 262–273. Springer, Heidelberg (2004)
5. Cai, J., Candes, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization **20**(4), 1956–1982 (2008)
6. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? Journal of the ACM **58**(3), 11:1–11:37 (2011)
7. Candès, E.J., Tao, T.: The power of convex relaxation: near-optimal matrix completion. IEEE Transactions on Information Theory **56**(5), 2053–2080 (2010)
8. Chatterjee, A., Govindu, V.M.: Efficient and robust large-scale rotation averaging. In: International Conference on Computer Vision, pp. 521–528 (2013)
9. Govindu, V.M.: Combining two-view constraints for motion estimation. In: Conference on Computer Vision and Pattern Recognition, pp. 218–225 (2001)
10. Hampel, F., Rousseeuw, P., Ronchetti, E., Stahel, W.: Robust Statistics: the Approach Based on Influence Functions, 2nd edn. John Wiley & Sons (1986)
11. Hartley, R., Aftab, K., Trumpf, J.: L1 rotation averaging using the Weiszfeld algorithm. In: Conference on Computer Vision and Pattern Recognition pp. 3041–3048 (2011)
12. Hartley, R.I., Trumpf, J., Dai, Y., Li, H.: Rotation averaging. International Journal of Computer Vision **103**, 267–305 (2013)
13. He, J., Balzano, L., Szlam, A.: Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video. In: Conference on Computer Vision and Pattern Recognition, pp. 1568–1575 (2012)
14. Keshavan, R.H., Montanari, A., Oh, S.: Matrix completion from a few entries. IEEE Transactions on Information Theory **56**(6), 2980–2998 (2010)
15. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
16. Mosteller, F., Tukey, J.: Data Analysis and Regression: A Second Course in Statistics. Addison-Wesley series in behavioral science. Addison-Wesley (1977)
17. Moulon, P., Monasse, P., Marlet, R.: Global Fusion of Relative Motions for Robust, Accurate and Scalable Structure from Motion. In: International Conference on Computer Vision, pp. 1568–1575 (2013)
18. Olsson, C., Enqvist, O.: Stable structure from motion for unordered image collections. In: Heyden, A., Kahl, F. (eds.) SCIA 2011. LNCS, vol. 6688, pp. 524–535. Springer, Heidelberg (2011)
19. Ozyesil, O., Singer, A., Basri, R.: Stable camera motion estimation using convex programming. SIAM Journal on Imaging Sciences **8**(2), 1220–1262 (2015)
20. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3D. ACM Transactions on Graphics **25**(3), 835–846 (2006)
21. Strecha, C., von Hansen, W., Gool, L.J.V., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
22. Toldo, R., Gherardi, R., Farenzena, M., Fusiello, A.: Hierarchical structure-and-motion recovery from uncalibrated images. Computer Vision and Image Understanding (2015)

23. Waters, A.E., Sankaranarayanan, A.C., Baraniuk, R.G.: SpaRCS: recovering low-rank and sparse matrices from compressive measurements. In: Neural Information Processing Systems, pp. 1089–1097 (2011)
24. Wilson, K., Snavely, N.: Robust global translations with 1DSfM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 61–75. Springer, Heidelberg (2014)
25. Zhou, T., Tao, D.: GoDec: randomized low-rank & sparse matrix decomposition in noisy case. In: International Conference on Machine Learning, pp. 33–40 (2011)