Spectral Motion Synchronization in SE(3)

Federica Arrigoni, Andrea Fusiello
DIEGM - University of Udine
Via delle Scienze, 208 - Udine (Italy)

Beatrice Rossi
AST Lab - STMicroelectronics
Via Olivetti, 2 - Agrate Brianza (Italy)

Abstract

This paper addresses the problem of motion synchronization (or averaging) and describes a simple, closed-form solution based on a spectral decomposition, which does not consider rotation and translation separately but works straight in SE(3), the manifold of rigid motions. Besides its theoretical interest, being the first closed form solution in SE(3), experimental results show that it compares favourably with the state of the art both in terms of precision and speed.

1. Introduction

In this paper we address the motion synchronization (a.k.a. motion averaging or motion registration) problem in the Special Euclidean Group, SE(3), which consists in recovering \( n \) absolute motions, i.e. rigid 3D displacements expressed in an absolute (external) coordinate system, starting from a redundant set of relative (pairwise) motions. This problem appears in the context of structure-from-motion (SfM) – where the absolute motions represent orientations and positions of cameras capturing a 3D scene, and multiple point-set registration – that requires to find the rigid transformations that bring multiple 3D point-sets into alignment.

In the literature on multiple point-set registration, the origins of motion synchronization can be traced back to the frame space methods [23] that optimize the internal coherence of the network of rotations and translations applied to the local coordinate frames, as opposed to solutions that optimize a cost function depending on the distance of corresponding points (e.g. [22, 6]).

In the structure-from-motion literature, global methods, that first solve for the motion by optimizing the network of relative motions and leave the 3D structure recovery at the end, are fairly recent (e.g. [19]), although the origins of these approaches can be traced back to [10].

Almost all the techniques address the motion synchronization problem by breaking it up into rotation and translation, and solving the two synchronization problems separately.

For what regards rotation synchronization, a theoretical analysis of the problem is reported in [15]. The absolute rotations can be recovered by using the quaternion representation of \( SO(3) \), as done in [10], or by distributing the error over cycles in the graph of neighbouring views [23]. In [14] a cost based on the \( \ell_1 \)-norm is used to average relative rotations, where each absolute rotation is updated in turn using the Weiszfeld algorithm. Martinec in [19] casts the problem as the optimization of an objective function based on the \( \ell_2 \)-norm of the compatibility error between relative estimates and unknown absolute orientations, and this approach is extended in [2] where approximate solutions are computed either via spectral decomposition or semidefinite programming. The sum of unsquared deviations is proposed in [28] as a more robust self consistency error. Chatterjee in [8] exploits the Lie-group structure of rotations, and combines an \( \ell_1 \) averaging in the tangent space with an iteratively reweighted least squares (IRLS) approach. In [4] the rotation synchronization problem is reformulated in terms of low-rank and sparse matrix decomposition.

As for translation synchronization methods, a discriminating factor relevant to our analysis is whether they use only relative motion information, or, in addition, exploit image point correspondences (e.g. [19, 29]). Let us focus on the former, which are more similar to our SE(3) approach. In [10] absolute positions are initialized as the least squares solution of a linear system of equations in the pairwise directions and orientations, and they are then improved through IRLS. In [7] a fast spectral solution to translation synchronization is proposed by reformulating the problem in terms of graph embedding. The method presented in [21] first computes pairwise directions through a robust subspace estimation and then derives absolute translations using a semidefinite relaxation. In [18] a linear solution which minimizes a geometric error in camera triplets is presented, while in [20] relative translations are computed through an a-contrario trifocal tensor estimation, and then absolute positions are recovered by using an \( \ell_\infty \) formulation.

A different approach for motion synchronization is followed in [11] where rotations and translations are jointly
considered. This method exploits the Lie-group structure of $SE(3)$ and uses an iterative scheme in which at each step the absolute motions are approximated by averaging relative motions in tangent space. In [12] robustness is introduced through random sampling in the measurement graph. Originally proposed in the SfM framework, this technique was also applied to multiple point-set registration [13] and simultaneous localization and mapping (SLAM) [1], where additional information about data reliability is introduced in the form of covariance matrices.

In this paper we propose a novel method for synchronizing relative motions in $SE(3)$, based on computing the least four eigenvectors of a $4n \times 4n$ matrix. Our approach is the first one that works in $SE(3)$ and has a closed-form solution, being based on a spectral decomposition. It can be seen as the extension to $SE(3)$ of the spectral synchronization proposed in [24] for $SO(2)$ and generalized in [25, 2] to $SO(3)$.

The simple matrix formulation of our method leads immediately to a weighted formulation, in much the same way as [28] did for $SO(N)$, that allows to embed it into an IRLS scheme in order to handle rogue measurements.

Experimental results on synthetic and real data show that it compares favourably with the state of the art both in terms of accuracy and efficiency.

2. Our Method

The motion synchronization problem consists in recovering $n$ absolute motions, i.e. rigid displacements in $\mathbb{R}^3$ expressed in an absolute (external) coordinate system, starting from a redundant set of relative (pairwise) motions. Such relative information is usually corrupted by a diffuse noise, in addition to sparse gross errors (outliers). Let $\mathcal{E} \subseteq \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$ denote the set of available pairs, which can be viewed as the set of edges of an undirected finite simple graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where vertices in $\mathcal{V}$ correspond to absolute motions. In practical applications this graph is far from complete, due to the lack of overlap between some pairs of images/scans. However, there is a significant level of redundancy among relative motions in general datasets, which can be used to distribute the error over all the nodes, avoiding drift in the solution.

Each motion can be viewed as an element of the Special Euclidean Group $SE(3)$, which is the semi-direct product of the Special Orthogonal Group $SO(3)$ with $\mathbb{R}^3$. As a matrix group, $SE(3)$ is a subgroup of the General Linear Group $GL(4)$, thus the inverse of a displacement and composition of displacements reduce to matrix operations. Accordingly, each absolute motion is described by a homogeneous transformation

$$M_i = \begin{pmatrix} R_i & t_i \\ 0 & 1 \end{pmatrix} \in SE(3)$$

where $R_i \in SO(3)$ and $t_i \in \mathbb{R}^3$ represent the rotation and translation components of the $i$-th transformation. Similarly, each relative motion can be expressed as

$$M_{ij} = \begin{pmatrix} R_{ij} & t_{ij} \\ 0 & 1 \end{pmatrix} \in SE(3)$$

where $R_{ij} \in SO(3)$ and $t_{ij} \in \mathbb{R}^3$ encode the transformation between frames $i$ and $j$. The link between absolute and relative motions is encoded by the compatibility constraint

$$M_{ij} = M_iM_j^{-1}$$

which is equivalent to $R_{ij} = R_iR_j^T$ and $t_{ij} = -R_iR_j^Tt_j + t_i$ by considering separately the rotation and translation terms. Relative motions can be seen as measurements for the ratios of the unknown group elements. Finding group elements from noisy measurements of their ratios is also known as the synchronization problem [9, 24].

The remainder of this section is organized as follows. In Sec. 2.1 we describe properties that hold when all the relative information is exact, necessary to define our technique. Then we derive our spectral solution to motion synchronization (Sec. 2.2). In Sec. 2.3 our method is embedded into an IRLS framework in order to handle outliers among relative motions. Finally, Sec. 2.4 briefly presents the extension of our method in $SE(N)$.

2.1. The Exact Case

The absolute transformations can be recovered from (3) – up to a global motion – if we express it in a useful equivalent way that takes into account all the relative information at once. For simplicity of exposition, we first consider the case where all the pairwise motions are available.

Let $X \in \mathbb{R}^{4n \times 4n}$ denote the block-matrix containing the ideal (noise free) relative motions and let $M \in \mathbb{R}^{4n \times 4}$ be the stack of the absolute motions, namely

$$M = \begin{bmatrix} M_1 \\ \vdots \\ M_n \end{bmatrix}, \quad X = \begin{bmatrix} I_4 & M_{12} & \ldots & M_{1n} \\ M_{21} & I_4 & \ldots & M_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n1} & M_{n2} & \ldots & I_4 \end{bmatrix}$$

where $I_4$ indicates the $4 \times 4$ identity matrix. If $M^{-b} \in \mathbb{R}^{4 \times 4n}$ is the concatenation of the inverse of absolute motions, i.e. $M^{-b} = [M_1^{-1} \ M_2^{-1} \ \ldots \ \ M_n^{-1}]$, then the compatibility constraint turns into $X = MM^{-b}$, and hence $\text{rank}(X) = 4$. Note that here $X$ is not symmetric positive semidefinite, in contrast to the case of $SO(3)$. Since $M^{-b} = nI_4$, we obtain

$$XM = nM$$

which means that – in the absence of noise – the columns of $M$ are 4 (independent) eigenvectors of $X$ associated to the
eigenvalue \( n \). Equation (5) is equivalent to
\[
(nI_n - X)M = 0. \tag{6}
\]
Thus the columns of \( M \) are a basis for the 4-dimensional null-space of \( L = (nI_n - X) \). The matrix \( L \) resembles a block Laplacian, as it will we clarified ahead.

Conversely, any basis \( U \) for \( \text{null}(L) \) will not coincide with \( M \) in general, since it will not be composed of euclidean motions. Specifically, it will not coincide with \([0\ 0\ 0\ 1]\) in every fourth row. In order to recover \( M \) from \( U \) it is sufficient to choose a different basis for \( \text{null}(L) \) that satisfies such constraint, which can be found by taking a suitable linear combination of the columns of \( U \). More precisely, let \( P \in \mathbb{R}^{n \times 4n} \) be the 0-1 matrix such that \( PU \in \mathbb{R}^{n \times 4} \) consists of the rows of \( U \) with indices multiple of four. The coefficient \( \alpha, \beta \in \mathbb{R}^4 \) of the linear combination are solution of
\[
PU\alpha = 0, \quad PU\beta = 1 \tag{7}
\]
where the first equation has a three-dimensional solution space. Let \( \alpha_1, \alpha_2, \alpha_3 \) be a basis for the null-space of \( PU \). Thus the columns of \( M \) corresponding to rotations coincide (up to a permutation) with \([U\alpha_1, U\alpha_2, U\alpha_3]\) and \( M \) is recovered as
\[
M = U[\alpha_1, \alpha_2, \alpha_3, \beta].
\]

Note that this post-processing on the eigenvectors is not required for the spectral method in \( SO(3) \), since any orthogonal basis for the null-space of \( L \) coincides (up to a permutation) with the stack of the absolute rotations.

We now consider the case of missing data, in which the graph \( \mathcal{G} \) is not complete. In this situation missing pairwise motions correspond to zero blocks in \( X \). Let \( A \in \mathbb{R}^{n \times n} \) be the adjacency matrix of \( \mathcal{G} \) and let \( D \in \mathbb{R}^{n \times n} \) be the degree matrix of \( \mathcal{G} \), i.e. the diagonal matrix that contains the degree of node \( i \) in its entry \( D_{i,i} \). It can be seen that Eq. (6) generalizes to
\[
((D - A) \otimes I_{4 \times 4}) \circ X)M = 0 \tag{8}
\]
where \( \otimes \) denotes the Kronecker product and \( \circ \) denotes the Hadamard product. The matrix \((D - A)\) is the Laplacian matrix of the graph \( \mathcal{G} \), which gets “inflated” to a \( 4 \times 4 \)-block structure by the Kronecker product with \( I_{4 \times 4} \) (a matrix filled by ones), and then is multiplied entry-wise with \( X \). Thus the columns of \( M \) are a basis for the 4-dimensional null-space of \( L = ((D - A) \otimes I_{4 \times 4}) \circ X \).

If \( \mathcal{G} \) is complete then \( D = (n-1)I_n \) and \( A = I_{n \times n} - I_n \), hence the matrix \( L \) reduces to the previous one.

Weighted graph.

In some applications we are given non-negative weights \( w_{ij} \) that reflect the reliability of the pairwise measurements. In other words, \( \mathcal{G} \) is a weighted graph with real weights, stored in the the symmetric adjacency matrix \( A = [w_{ij}] \). Accordingly, the degree matrix \( D \) of the weighted graph is defined as \( D_{i,i} = \sum_{k \neq i} w_{ik} \). Equation (8) still holds with these definitions, thus our spectral method extends to weighted motion synchronization.

2.2. Dealing with Noise

We now consider the case where the pairwise motions are corrupted by noise, hence they do not satisfy equations (3) and (8) exactly. Thus the goal is to recover the absolute motions such that they are “maximally compatible” with the available relative information. In order to address this motion synchronization problem, we consider an algebraic cost function that measures the residuals (in the Frobenius norm sense) of Equation (8), namely
\[
\min_{M \in \mathbb{R}^{n \times 4}} \| \hat{L}M \|_F^2 \tag{9}
\]
with the additional constraint \( \| \mathbf{m}_4 \|_F = c \) in order to fix the global scale. Here \( \mathbf{m}_4 \) denotes the fourth column of \( M \), \( \hat{X} \) denotes a noisy version of the ideal matrix \( X \), which contains the measured relative motions \( \hat{M}_{ij} \in SE(3) \), and \( \hat{L} = ((D - A) \otimes I_{4 \times 4}) \circ \hat{X} \). Hereafter we will consistently use the hat accent to denote noisy measurements. Such a problem is difficult to solve since the feasible set \( SE(3)^n \times \cdots \times SE(3) \) is non-convex.

In order to make the computation tractable, we do not solve Problem (9) directly, but we proceed as follows. First, we look for an orthogonal basis for the (approximated) 4-dimensional null-space of \( \hat{L} \), by solving the following optimization problem
\[
\min_{U^\top U = nI_4} \| \hat{L} U \|_F^2. \tag{10}
\]
In other words, we solve the homogeneous system of equations \( \hat{L} U = 0 \) in the least-squares sense, where the solution space is known to have approximately dimension 4.

Then, we find an estimate for \( M \) within this space by forcing the solution to coincide with \([0\ 0\ 0\ 1]\) in every fourth row. Finally, we project in \( SO(3) \) all the 3 \( \times \) 3 blocks corresponding to rotations by using Singular Value Decomposition (SVD).

**Proposition 1.** Problem (10) admits a closed-form solution, which is given by the 4 eigenvectors of \( \hat{L}^\top \hat{L} \) associated to the 4 smallest eigenvalues.

**Proof.** We first observe that Problem (10) coincides with
\[
\min_{U^\top U = nI_4} \text{tr}(U^\top (\hat{L}^\top \hat{L})U). \tag{11}
\]
Let \( \mathcal{F} \) be the unconstrained cost function corresponding to this problem, namely
\[
\mathcal{F}(U) = \text{tr}(U^\top (\hat{L}^\top \hat{L})U) + \text{tr}(A(U^\top U - nI_4)) \tag{12}
\]
where $A \in \mathbb{R}^{4 \times 4}$ is a symmetric matrix of unknown Lagrange multipliers. Setting to zero the partial derivatives of $\mathcal{F}$ with respect to $U$ we obtain

$$\frac{\partial \mathcal{F}}{\partial U} = 2(\hat{L}^T\hat{L})U + 2UA = 0 \Rightarrow (\hat{L}^T\hat{L})U = -UA. \quad (13)$$

Let $u_i$ be any four eigenvectors of $\hat{L}^T\hat{L}$ (normalized so that $\|u_i\| = \sqrt{\lambda_i}$) and let $\lambda_i$ be the corresponding eigenvalues. Then $U = [u_1 u_2 u_3 u_4]$ satisfies both (13) and the constraint $U^TU = nI_4$, with $A = -\text{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ (indeed $\hat{L}^T\hat{L}$ admits an orthonormal basis of real eigenvectors since it is symmetric). In other words, any quadruple of eigenvectors is a stationary point for the objective function $\mathcal{F}$. The minimum is attained in (11) if $u_i$ are the 4 least eigenvectors of $\hat{L}^T\hat{L}$.

Proposition 1 guarantees that the solution to problem (10) is given by the 4 least eigenvectors of $\hat{L}^T\hat{L}$, which coincide with the 4 least right singular vectors in the Singular Value Decomposition (SVD) of $\hat{L}$. Such a solution represents the best (in the Frobenius norm sense) 4-dimensional approximation for null($\hat{L}$). Within such a space, we find the solution that is closest to have every fourth row equal to $[0 0 0 1]$ by solving system (7) in the least-squares sense. Then, such a solution is projected onto $SE(3)^n$ -- as in [5] -- by forcing every fourth row to $[0 0 0 1]$ and projecting $3 \times 3$ rotation blocks onto $SO(3)$ through SVD.

This technique has the advantage of being extremely fast, as motion synchronization is cast to eigenvalue decomposition of a $4n \times 4n$ matrix. Moreover, in practical application the measurement graph $G$ is sparse, thus employing sparse eigen-solvers (such as MATLAB \texttt{eigs}) increases its efficiency. From the computational complexity point of view, the Lanczos method (implemented by \texttt{eigs}) is “nearly linear” since every iteration is linear in $n$, if the matrix is sparse, but the number of iterations is not constant.

### 2.3. Dealing with Outliers

The fact that our spectral method copes easily with weights on individual relative motions allows a straightforward extension to gain resilience to rogue input measures via Iteratively Reweighted Least Squares (IRLS).

First, we solve (10) to obtain an estimate for $M$ with given weights$^1$ as explained in the previous section, then we update the weights using the current estimate of absolute motions, and these steps are iterated until convergence. In our experiments we used the Cauchy weight function [16]

$$w_{ij} = \frac{1}{1 + \left(\frac{x_i}{c}\right)^2}, \quad (14)$$

where $r_{ij} = \|\hat{M}_{ij} - M_iM_j^{-1}\|_F$. The tuning constant $c$ has been chosen, as customary, based on the median absolute deviation (MAD): $c = 1.482 \theta \text{med}(|r - \text{med}(r)|)$, where $\text{med}(\cdot)$ is the median operator, $r$ is the vectorization of the residuals $r_{ij}$, and $\theta = 2$.

### 2.4. Generalization to SE(N)

In this paper we focus on $SE(3)$ because this group arises in several applications. However, it is straightforward to see that our analysis and the derived spectral method apply equally well to any dimension.

Suppose that we are given a redundant number of pairwise ratios $M_{ij} \in SE(N)$, and we want to estimate the associated group elements $M_i \in SE(N)$, which represent rigid displacements in $\mathbb{R}^N$. If the graph is complete then -- in the absence of noise -- the block-matrix $X \in \mathbb{R}^{(N+1)n \times (N+1)n}$ has rank $N + 1$, and the columns of $M$ are $N + 1$ eigenvectors of $X$ with eigenvalue $n$. If the graph is not complete then Equation (8) still hold, and hence the columns of $M$ form a basis for the $(N + 1)$-dimensional null-space of $L$. Thus we can generalize our spectral method to synchronize elements of $SE(N)$, by computing the $N + 1$ least eigenvectors of $\hat{L}^T\hat{L}$.

### 3. Experiments

In this section we evaluate our spectral method – henceforth called EIG-SE(3) – on both simulated and real data in terms of accuracy, execution cost and robustness to outliers. We compare EIG-SE(3) to several techniques from the state-of-the-art. All the experiments are performed in MATLAB on a MacBook Air with i5 dual-core @ 1.3 GHz. In order to compare estimated and ground-truth absolute motions, we find the optimal transformation that aligns them by applying single averaging [14] for the rotation term and least-squares for the scale and translation. We use the angular distance and Euclidean norm to measure the accuracy of absolute rotations and translations respectively.

#### 3.1. Simulated Data

In these experiments we consider $n$ absolute motions in which rotations are sampled from random Euler angles and translation components follow a standard Gaussian distribution. The level of sparsity of the measurement graph is defined through the average degree $d$ of nodes. The available pairwise motions are corrupted by a multiplicative noise, where the rotation component has axis uniformly distributed over the unit sphere and angle following a Gaussian distribution with zero mean and standard deviation $\sigma_R \in [1^\circ, 10^\circ]$, and the translation components are sampled from a Gaussian distribution with zero mean and standard deviation $\sigma_T \in [0.01, 0.1]$. In this way we perturb both direction and magnitude of pairwise translations. All the results are averaged over 50 trials.

$^1$The initial weights are all 1 by default, but they can be initialized from any reliability information coming from the relative motion estimation procedure.
We evaluate the effect of noise on rotations and translations both separately and together, by considering \( n = 100 \) absolute motions, in the cases \( d = 5 \) and \( d = 30 \), which correspond to about 95% and 70% of missing pairs, respectively. Higher values of \( d \) correspond to better conditioned problems, with the same qualitative behaviour as \( d = 30 \). Please note that in the real cases reported in Tab. 3, the percentage of missing pairs ranges from 30% to 90%.

**Rotation synchronization.** As for rotations, besides Govindu-SE(3) [11], we consider general synchronization techniques such as the Weiszfeld algorithm [14], spectral relaxation [2] (EIG), semidefinite programming [2] (SDP), the L1-IRLS algorithm [8], and the R-GoDec algorithm [4]. Methods based on quaternions (such as [10]) have been already proved inferior to the other methods in [19]. The code of L1-IRLS is available on-line, while in the other cases we use our implementation. In this simulation we do not perturb the relative rotations \((\sigma_R = 0)\), thus all the methods are given ground-truth relative/absolute rotations. Noise on rotational component influences also the translation errors with results qualitatively similar to those reported here.

Figure 1 reports the mean angular errors on the absolute rotations as a function of \( \sigma_R \), obtained by running the rotation synchronization techniques mentioned above. The best accuracy is obtained by EIG-SE(3) together with EIG, SDP and Govindu-SE(3). On the contrary, the robust approaches R-GoDec, L1-IRLS and Weiszfeld yield worse results, to different extents, because they inherently trade robustness for statistical efficiency.

The noise on relative translations does not have any influence on absolute rotations, hence the value of \( \sigma_T \) is meaningless in this experiment.

**Translation synchronization.** As for translations, we consider only methods working in frame space, i.e. not requiring point correspondences, such as SDR [21], the graph-embedding approach by Brand et al. [7] and the works of Govindu [10, 11]. Among these methods, only EIG-SE(3) and Govindu-SE(3) [11] are influenced by the noise on the translation norms, for they work in SE(3), while this does not influence the remaining algorithms, which take as input relative translation directions. The code of SDR is available on-line, while in the other cases we use our implementation. In this simulation we do not perturb the relative rotations \((\sigma_R = 0)\), thus all the methods are given ground-truth relative/absolute rotations. Noise on rotational component influences also the translation errors with results qualitatively similar to those reported here.

Figure 2 shows the mean errors on the absolute translations as a function of \( \sigma_T \) (units are commensurate with the simulated data), obtained by running the techniques mentioned above. Both EIG-SE(3) and Govindu-SE(3) outperform all the analysed methods in terms of accuracy.

When the measurement graph is extremely sparse \((d = 5)\) the methods by Govindu [10] and Brand et al. [7] yield larger errors than usual; by inspecting the solution it is found that this corresponds to wrong solutions concentrated...
In this experiment we consider motion synchronization. We also analysed the execution time of motion synchronization, by varying the number of absolute motions from $n = 100$ to $n = 1000$, all the others parameters being fixed. More precisely, we choose the values $d = 10$, $\sigma_T = 0.05$ and $\sigma_R = 5^c$ to define sparsity and noise. Figure 5 reports the running times of the analysed algorithms as a function of the number of nodes in the measurements graph, showing that EIG-SE(3) is remarkably faster than Govindu-SE(3) and SDR. Indeed SDR solves a semidefinite programming problem and Govindu-SE(3) uses an iterative approach in which absolute motions are updated by performing multiple averaging in the tangent space; both these operations are more expensive than computing the least four eigenvectors of a $4n \times 4n$ matrix.

The rundown of these experiments is that, EIG-SE(3) achieves the same optimal accuracy of its closest competitor [11] in considerably less time.

Outliers influence. In this experiment we study the resilience to outliers of EIG-SE(3) with IRLS. We consider $n = 100$ absolute motions sampled as before and we fix $d = 30$ to define sparsity. Since we are interested in analysing exact recovery in the presence of outliers, noise is not introduced in this simulation. The fraction of wrong relative motions – randomly generated – varies from 10% to 50%. Figure 6 reports the mean errors obtained by EIG-SE(3) and its IRLS modification: the empirical breakdown point of EIG-SE(3) + IRLS is about 45%.
provide results obtained by using ground-truth scales, in ad-
the specific method for computing the scales, and we also
are fully specified. For this reason we are agnostic about
point-set registration problem, where the relative motions
motion problem, which is not shared, e.g., by the multiple
this indeterminacy is an idiosyncrasy of the structure from
synchronization task, strictly speaking. As a matter of fact,
confirming that EIG-SE(3) provides a good starting point
chronization step takes about 1s for the largest sequences.
If we concentrate on the EIG-SE(3)-GT columns, we can
see that it achieves the optimum before BA in most datasets,
confirming the effectiveness of our method for synchron-
izing relative motions, when the latter are fully specified.
Without ground-truth scales, good estimates of motion pa-
rameters are still obtained, and precision increases by using
MCB rather than the iterative approach. The error after BA
is always very small and almost equal to the other methods,
confirming that EIG-SE(3) provides a good starting point
for bundle adjustment.

**Large-scale Datasets.** We test our technique on irregular
large-scale collections of images taken from [29], for which
recovering camera orientations/locations is challenging.
Since our MATLAB implementation of Horton’s Algo-

Figure 6. Mean errors on the absolute motions versus outliers con-
tamination.

**3.2. Real Data**

We apply EIG-SE(3) with IRLS to the structure from
motion problem, considering both the EPFL benchmark
[27] and unstructured, large-scale image sequences from
[29]. The latter are available on-line together with the rela-
tive motions, while for the EPFL benchmark we computed
them following a standard approach based on the essential
matrix with a final bundle adjustment (BA) refinement of
camera pairs.

Owing to the depth-speed ambiguity, the magnitude of
relative translations (also referred to as *epipolar scales*) are
undefined. Therefore, the input relative motions do not fully
specify elements of $SE(3)$, and the unknown scales have to
be computed.

A straightforward approach (suggested in [11]) consists
in iteratively updating these epipolar scales, i.e. during each
iteration the scale of the translation of $\hat{M}_{ij}$ is set equal to
that of $M_i M_j^{-1}$, where $M_i$ and $M_j$ are the current estimates
of camera motions. The starting scales are all equal to 1 and
the procedure is iterated until convergence. In our imple-
mentation this is combined with IRLS in the same loop: in
one step we update the IRLS weights and in the next step
we update the epipolar scales.

A different approach is proposed in [3], where a two-
stage method is developed for computing the epipolar scales
based on the knowledge of two-view geometries only. First,
a Minimum Cycle Basis (MCB) for the measurement graph
is extracted by using Horton’s Algorithm [17], then all the
scales are recovered simultaneously by solving a homoge-
nous linear system. This approach is based on the observa-
tion that the compatibility constraints associated to these
cycles can be seen as equations in the unknown scales. In
this way all the epipolar scales are computed before per-
forming motion synchronization.

However, computing the epipolar scales is not part of
the synchronization task, strictly speaking. As a matter of fact,
this indeterminacy is an idiosyncrasy of the structure from
motion problem, which is not shared, e.g., by the multiple
point-set registration problem, where the relative motions
are fully specified. For this reason we are agnostic about
the specific method for computing the scales, and we also
provide results obtained by using ground-truth scales, in ad-
tition to the approaches mentioned above.

**EPFL Benchmark.** The EPFL Benchmark datasets [27]
contain from 8 to 30 images, and provide ground-truth ab-
solute motions.

Results are reported in Tab. 1 and Tab. 2, which show the
mean errors of motion synchronization before and after ap-
plying a two-step bundle adjustment, as done in [20], where
in the first step rotations are kept fixed.

We consider three versions of EIG-SE(3), which differ
for the technique chosen to recover the epipolar scales,

namely using ground-truth scales (GT), computing scales
through [3] (MCB), and updating scales iteratively (Iter).
Our spectral solution is compared with the global SfM
pipeline described by Moulon et al. [20] and by Ozyesil
et al. [21]. We also consider the pipeline obtained by combi-
ning the rotation synchronization technique in [4] with the
translation synchronization method in [7]. As a reference,
we included in the comparison the sequential SfM pipeline
**Bundler** [26].

With the exception of Moulon et al. and **Bundler**, for
which results are taken from [20], all the other methods are
given the same relative motions as inputs.

Both EIG-SE(3) and all the analysed techniques achieve
a high precision, obtaining an average rotation error less
than 0.1 degrees and an average translation error of the or-
der of millimetres, after the final BA. Our method is able to
recover camera parameters efficiently, since the motion syn-
chronization step takes about 1s for the largest sequences.

If we concentrate on the EIG-SE(3)-GT columns, we can
see that it achieves the optimum before BA in most datasets,
confirming the effectiveness of our method for synchron-
izing relative motions, when the latter are fully specified.
Without ground-truth scales, good estimates of motion pa-
rameters are still obtained, and precision increases by using
MCB rather than the iterative approach. The error after BA
is always very small and almost equal to the other methods,
confirming that EIG-SE(3) provides a good starting point
for bundle adjustment.
Table 1. Mean angular errors (degrees) on camera rotations for the EPFL benchmark. Moulon et al. is missing in this table because rotation errors are not reported in [20].

| Dataset         | EIG-SE(3)-GT | EIG-SE(3)-Iter | EIG-SE(3)-MCB | Ozyesil et al. | R-GoDec+Brand et al. |
|-----------------|--------------|----------------|----------------|----------------|----------------------|
|                 | pre BA       | post BA        | pre BA         | post BA        | pre BA              | post BA              |
| HerzJesuP8      | 0.04         | 0.03           | 0.03           | 0.03           | 0.03                | 0.06                 |
| HerzJesuP25     | 0.06         | 0.03           | 0.06           | 0.04           | 0.06                | 0.14                 |
| FountainP11     | 0.03         | 0.03           | 0.03           | 0.04           | 0.04                | 0.03                 |
| EntryP10        | 0.04         | 0.02           | 0.10           | 0.02           | 0.11                | 0.56                 |
| CastleP19       | 1.48         | 0.06           | 1.48           | 0.06           | 2.46                | 3.69                 |
| CastleP30       | 0.53         | 0.05           | 0.47           | 0.05           | 0.77                | 1.97                 |

Table 2. Mean errors (meters) on camera translations for the EPFL benchmark.

| Dataset         | EIG-SE(3)-GT | EIG-SE(3)-Iter | EIG-SE(3)-MCB | Ozyesil et al. | R-GoDec+Brand et al. | Moulon et al. | BUNDLER |
|-----------------|--------------|----------------|----------------|----------------|----------------------|---------------|---------|
|                 | pre BA       | post BA        | pre BA         | post BA        | pre BA              | post BA        | pre BA  |
| HerzJesuP8      | 0.004        | 0.004          | 0.659          | 0.004          | 0.007               | 0.009          | 0.004 |
| HerzJesuP25     | 0.008        | 0.008          | 1.152          | 0.022          | 0.065               | 0.009          | 0.005 |
| FountainP11     | 0.004        | 0.003          | 0.236          | 0.003          | 0.004               | 0.003          | 0.003 |
| EntryP10        | 0.009        | 0.008          | 0.309          | 0.008          | 0.349               | 0.009          | 0.006 |
| CastleP19       | 0.709        | 0.034          | 4.986          | 0.034          | 3.967               | 0.035          | 1.769 |
| CastleP30       | 0.212        | 0.032          | 1.974          | 0.035          | 3.866               | 0.034          | 1.393 |

Results are reported in Tab. 3, which shows the median errors of motion synchronization before applying bundle adjustment. We also report the number of cameras reconstructed and the percentage of missing pairs, which refer to the largest parallel-rigid subgraph, extracted as explained in [21]. The results of 1DSfM are taken from [29], where rotation errors are not analysed. EIG-SE(3) with iterative scale estimate performs equal or better than 1DSfM in 7 cases out of 11, and it recovers camera rotations accurately.

Computation times of EIG-SE(3) (MATLAB implementation on a MacBook Air with i5 dual-core @ 1.3 GHz) reported in Tab. 3 are hardly comparable with those reported in [29], as they refer to a compiled code on a much powerful computer. However, if we can assume that the speed gain from MATLAB to C++ (for non-trivial algorithms) is at least 10 times, as common wisdom suggests, we might then conjecture that EIG-SE(3) implemented in C++ would compare favourably with 1DSfM. Moreover, performing parallel computation for updating scales/weights could further improve its computational efficiency.

The rundown of these experiments with real datasets shows that, endowed with IRLS to withstand outliers and combined with a method for estimating the unknown epipolar scales, EIG-SE(3) can compete with state-of-the-art global pipelines.

4. Conclusion

We presented a new closed-form method for motion synchronization in $SE(3)$. The method is fast and simple, being based on a spectral decomposition, and theoretically relevant, for it works in the manifold of rigid motions.

Table 3. Median errors (rotation in degrees, translation in metres) on the datasets from [29] before BA. Boldface denotes the lowest translation error. Times are in minutes.

| Dataset         | n  | miss % | rot. | tra. | time | rot. | tra. |
|-----------------|----|--------|------|------|------|------|------|
| Roman Forum     | 1102| 89     | 2.1  | 13.5 | 16.3 | 6.1  | 3.5  |
| Vienna Cathedral| 898 | 75     | 1.6  | 7    | 17.6 | 6.6  | 5    |
| Alamo           | 606 | 50     | 1.3  | 1.5  | 14   | 1.1  | 2.6  |
| Notre Dame      | 553 | 32     | 0.8  | 0.5  | 14.7 | 10   | 2.6  |
| Tower of London | 489 | 81     | 2.8  | 7.3  | 3.0  | 11   | 1.3  |
| Montreal N. Dame| 467 | 53     | 0.6  | 0.8  | 6.3  | 2.5  | 1.9  |
| Yorkminster     | 448 | 73     | 1.9  | 7.2  | 3.2  | 10.7 | 2.6  |
| Madrid Metropolis| 370 | 69    | 5.6  | 9.6  | 2.5  | 9.9  | 0.7  |
| NYC Library     | 358 | 71     | 3.1  | 2.5  | 2.2  | 2.5  | 1.3  |
| Piazza del Popolo| 345 | 60    | 0.9  | 1.6  | 2.2  | 3.1  | 1    |
| Ellis Island    | 240 | 33     | 0.8  | 2.8  | 1.8  | 3.7  | 0.5  |

Our experiments showed that our method: i) has the same accuracy as its closest competitor [11] but it is much faster, and ii) combined with a method for estimating the unknown translation norms, it can be profitably used in a global structure from motion pipeline with state of the art performances.

The MATLAB implementation of EIG-SE(3) will be made publicly available.

References

[1] M. Agrawal. A lie-algebraic approach for consistent pose registration for general euclidean motion. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1891–1897, 2006.
