Tensor-Based Synchronization and the Low-Rankness of the Block Trifocal Tensor

Daniel Miao * **Gilad Lerman*** **Joe Kileel**†

Abstract

The block tensor of trifocal tensors provides crucial geometric information on the three-view geometry of a scene. The underlying synchronization problem seeks to recover camera poses (locations and orientations up to a global transformation) from the block trifocal tensor. We establish an explicit Tucker factorization of this tensor, revealing a low multilinear rank of $(6,4,4)$ independent of the number of cameras under appropriate scaling conditions. We prove that this rank constraint provides sufficient information for camera recovery in the noiseless case. The constraint motivates a synchronization algorithm based on the higher-order singular value decomposition of the block trifocal tensor. Experimental comparisons with state-of-the-art global synchronization methods on real datasets demonstrate the potential of this algorithm for significantly improving location estimation accuracy. Overall this work suggests that higher-order interactions in synchronization problems can be exploited to improve performance, beyond the usual pairwise-based approaches.

1 Introduction

Synchronization is crucial for the success of many data-intensive applications, including structure from motion, simultaneous localization and mapping (SLAM), and community detection. This problem involves estimating global states from relative measurements between states. While many studies have explored synchronization in different contexts using pairwise measurements, few have considered measurements between three or more states. In real-world scenarios, relying solely on pairwise measurements often fails to capture the full complexity of the system. For instance, in networked systems, interactions frequently occur among groups of nodes, necessitating approaches that can handle higher-order relationships. Extending synchronization to consider measurements between three or more states, however, increases computational complexity and requires sophisticated mathematical models. Addressing these challenges is vital for advancing various technological fields. For example, higher-order synchronization can improve the accuracy of 3D reconstructions in structure from motion by leveraging more complex geometric relationships. In SLAM, it enhances mapping and localization precision in dynamic environments by considering multi-robot interactions. Similarly, in social networks, it could lead to more accurate identification of tightly-knit groups. Developing efficient algorithms to handle higher-order measurements will open new research avenues and make systems more resilient and accurate.

In this work, we focus on a specific instance of the synchronization problem within the context of structure from motion in 3D computer vision, where each state represents the orientation and location of a camera. Traditional approaches rely on relative measurements encoded by fundamental matrices, which describe the relative projective geometry between pairs of images. Instead, we consider higher-order relative measurements encoded in trifocal tensors, which capture the projective information between triplets of images. Trifocal tensors uniquely determine the geometry of three views, even in the collinear case [1], making them more favorable than triplets of fundamental matrices

*School of Mathematics, University of Minnesota (miao0022@umn.edu, lerman@umn.edu)

†Department of Mathematics and Oden Institute for Computational Engineering and Sciences, University of Texas at Austin (jkileel@math.utexas.edu)

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

for synchronization. To understand the structure and properties of trifocal tensors in multi-view geometry, we carefully study the mathematical properties of the block tensor of trifocal tensors. We then use these theoretical insights to develop effective synchronization algorithms.

Directly relevant previous works. In the structure from motion problem, synchronization has traditionally been done using incremental methods, such as Bundler [2] and COLMAP [3]. These methods process images sequentially, gradually recovering camera poses. However, the order of image processing can impact reconstruction quality, as error may significantly accumulate. Bundle adjustment [4], which jointly optimizes camera parameters and 3D points, has been used to limit drifting but is computationally expensive.

Alternatively, global synchronization methods have been proposed. These methods process multiple images simultaneously, avoiding iterative procedures and offering more rigorous and robust solutions. Global methods generally optimize noisy and corrupted measurements by exploiting the structure of relative measurements and imposing constraints. Many global methods solve for orientation and location separately, using structures on $SO(3)$ and the set of locations. Solutions for retrieving camera poses from pairwise measurements have been developed for camera orientations [5, 6, 7, 8, 9, 10], camera locations [11, 12, 13], and both simultaneously [14, 15, 16, 17]. Some methods explore the structure on fundamental or essential matrices [18, 19, 20].

Several attempts to extract information from trifocal tensors include works by: Leonardos et al. [21], which parameterizes calibrated trifocal tensors with non-collinear pinhole cameras as a quotient Riemannian manifold and uses the manifold structure to estimate individual trifocal tensors robustly; Larsson et al. [22], which proposes minimal solvers to determine calibrated radial trifocal tensors for use in an incremental pipeline, handling distorted images with constraints invariant to radial displacement; and Moulon et al. [23], which introduces a structure from motion pipeline, retrieving global rotations via cleaning the estimation graph and solving a least squares problem, and solving for translations by estimating trifocal tensors individually by linear programs. To our knowledge, no prior works develop a global pipeline where the synchronization operates directly on trifocal tensors.

Contribution of this work. The main contributions of this work are as follows:

- We establish an explicit Tucker factorization of the block trifocal tensor when its blocks are suitably scaled, demonstrating a low multilinear rank of $(6,4,4)$. Moreover, we prove that this rank constraint is sufficient to determine the scales and fully characterizes camera poses in the noiseless case.
- We develop a method for synchronizing trifocal tensors by enforcing this low rank constraint on the block tensor. We validate the effectiveness of our method through tests on several real datasets in structure from motion.

2 Low-rankness of the block trifocal tensor

We first briefly review relevant background material in Section 2.1. Then we present the main new construction and theoretical results in Section 2.2.

2.1 Background

2.1.1 Cameras and 3D geometry

Given a collection of n images $I_1,...,I_n$ of a 3D scene, let $t_i \in \mathbb{R}^3$ and $R_i \in SO(3)$ denote the location and orientation of the camera associated with the image I_i in the global coordinate system. Moreover, each camera is associated with a calibration matrix K_i that encodes the intrinsic parameters of a camera, including the focal length, the principal points, and the skew parameter. Then, the 3×4 camera matrix has the following form, $P_i = K_i R_i [I_{3\times 3}, -t_i]$ and is defined up to nonzero scale. Three-dimensional world points X are represented as $\mathbb{R}^4$ vectors in homogeneous coordinates, and the projection of X into the image corresponding to P is $x = PX$. 3D world lines L can be represented via Plücker coordinates as an $\mathbb{R}^6$ vector. Then the projection of L onto the image corresponding to P is $l = \mathcal{P}L$, where $\mathcal{P}$ is the 3×6 line projection matrix. It can be written as $\mathcal{P} = \left[P^2 \wedge P^3; P^3 \wedge P^1; P^1 \wedge P^2\right]$ where P^i is the i-th row of the camera matrix P and wedge denotes exterior product. Explicitly the (i,j) element of the line projection matrix can be calculated as the determinant of the submatrix, where the i-th row

is omitted and the column are selected as the j-th pair from [(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)]. The elements on the second row are multiplied by -1.

To retrieve global poses, relative measurement of pairs or triplets of images is needed. Let x_i and x_j be any pair of corresponding keypoints in images I_i and I_j respectively, meaning that they are images of a common world point. The fundamental matrix F_{ij} is a 3×3 matrix such that $x_i^T F_{ij} x_j = 0$. It is known that F_{ij} encodes the relative orientation $R_{ij} = R_i R_j^T$ and translation $t_{ij} = R_i(t_i - t_j)$ through $F_{ij} = K_i^{-T}[t_{ij}]_\times R_{ij} K_j^{-1}$. The essential matrix corresponds to the calibrated case, where $K_i = I_{3\times3}$ for all i.

2.1.2 Trifocal tensors

Analogous to the fundamental matrix, the trifocal tensor T_{ijk} is a $3\times3\times3$ tensor that relates the features across images and characterizes the relative pose between a triplet of cameras P_i,P_j,P_k. The trifocal tensor T_{ijk} corresponding to cameras P_i,P_j,P_k can be calculated by

$$(T_{ijk})_{wqr} = (-1)^{w+1}\det\begin{bmatrix}\sim P_i^w \\ P_j^q \\ P_k^r\end{bmatrix}, \tag{1}$$

where P_i^w is the w-th row of P_i, and $\sim P_i^w$ is the 2×4 submatrix of P_i omitting the w-th row. The trifocal tensor determines the geometry of three cameras up to a global projective ambiguity, or up to a scaled rigid transformation in the calibrated case. In addition to point correspondences, trifocal tensors satisfy constraints for corresponding lines, and mixtures thereof. For example, let l_i,l_j,l_k be corresponding image lines in the views of cameras P_i,P_j,P_k respectively, then the lines are related through the trifocal tensor T_{ijk} by $\left(l_j^T[(T_{ijk})_{1::},(T_{ijk})_{2::},(T_{ijk})_{3::}]l_k\right)[l]_\times = 0^T$, where $[l]_\times$ denotes the $\times3$ skew-symmetric matrix corresponding to cross product by l. We refer to [1] for more details of the properties of a trifocal tensor. We include the standard derivation of the trifocal tensor in Appendix A.1.

Since corresponding lines put constraints on the trifocal tensor, one advantage of incorporating trifocal tensors into structure from motion pipelines is that trifocal tensors can be estimated purely from line correspondences or a mixture of points and lines. Fundamental matrices can not be estimated directly from line correspondences, so the effectiveness of pairwise methods for datasets where feature points are scarce is limited. Furthermore, trifocal tensors have the potential to improve location estimation. From pairwise measurements, one can only get the relative direction but not the scale and the location estimation in the pairwise setting is a "notoriously difficult problem" (quoting from pages 316-317 of [24]). However, trifocal tensors encode the relative scales of the direction and can greatly simplify the location estimation procedure. We refer to several works on characterizing the complexity of minimal problems for individual trifocal tensors [25, 26], and on developing methods for solving certain minimal problems [27],[28], [29], [30], [31], [32], [33]. We also refer to [34] for a survey paper on structure from motion, which discusses minimal problem solvers from the perspective of computational algebraic geometry.

2.1.3 Tucker decomposition and the multilinear rank of tensors

We review basic material on the Tucker decomposition and the multilinear rank of a tensor. We refer to [35] for more details while adopting its notation. Let $T \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ be an order N tensor. The *mode-i flattening* (or *matricization*) $T_{(i)} \in \mathbb{R}^{I_i\times(I_1\ldots I_{i-1}I_{i+1}\ldots I_N)}$ is the rearrangement of T into a matrix by taking mode-i fibers to be columns of the flattened matrix. By convention, the ordering of the columns in the flattening follows lexicographic order of the modes excluding i. Symbols $\otimes$ and $\odot$ denote the Kronecker product and the Hadamard product respectively. The norm on tensors is defined as $\|T\| = \|T_{(1)}\|_F$. The i-rank of T is the column rank of $T_{(i)}$ and is denoted as $\text{rank}_i(T)$. Let R_i=$\text{rank}_i(T)$. Then the *multilinear rank* of T is defined as mlrank(T) = $(R_1,R_2,\ldots,R_N)$. The *i-mode product* of T with a matrix $U \in \mathbb{R}^{m\times I_i}$ is a tensor in $\mathbb{R}^{I_1\times\cdots\times I_{i-1}\times m\times I_{i+1}\times\cdots\times I_N}$ such that

$$(T\times_i U)_{j_1\cdots j_{i-1}kj_{i+1}\cdots j_N} = \sum_{j_i=1}^{I_i} T_{j_1j_2\cdots j_N}U_{kj_i}.$$

Then, the *Tucker decomposition* of $T \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ is a decomposition of the following form:

$$T = \mathcal{G}\times_1 A_1\times_2 A_2\times_3\cdots\times_N A_N = [\![\mathcal{G};A_1,A_2,\ldots,A_N]\!],$$

where $\mathcal{G} \in \mathbb{R}^{Q_1 \times \cdots \times Q_N}$ is the core tensor, and $A_n \in \mathbb{R}^{I_n \times Q_n}$ are the factor matrices. Without loss of generality, the factor matrices can be assumed to have orthonormal columns. Given the multilinear rank of the core tensor $(R_1,...,R_N)$, the Tucker decomposition approximation problem can be written as

$$\underset{\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}, A_i \in \mathbb{R}^{I_i \times R_i}}{\operatorname{argmin}} \|T - [\![\mathcal{G}; A_1, A_2, ..., A_N]\!]\|. \tag{2}$$

A standard way of solving (2) is the *higher-order singular value decomposition* (*HOSVD*). The HOSVD is computed with the following steps. First, for each i calculate the factor matrix A_i as the R_i leading left singular vectors of $T_{(i)}$. Second, set the core tensor $\mathcal{G}$ as $\mathcal{G} = T \times_1 A_1^T \times_2 \cdots \times_N A_N^T$. Though the solution from HOSVD will not be the optimal solution to (2), it satisfies a quasi-optimality property: if T^* is the optimal solution, and T' the solution from HOSVD, then

$$\|T - T'\| \leq \sqrt{N} \|T - T^*\|. \tag{3}$$

2.2 Low Tucker rank of the block trifocal tensor and one shot camera retrieval

Suppose we are given a set of camera matrices $\{P_i\}_{i=1}^n$ with $n \geq 3$ and scales fixed on each camera matrix. Define the *block trifocal tensor* T^n to be the $3n \times 3n \times 3n$ tensor, where the $3 \times 3 \times 3$ sized ijk block is the trifocal tensor corresponding to the triplet of cameras P_i, P_j, P_k. We assume for all blocks that have overlapping indices, the corresponding $3 \times 3 \times 3$ tensor is also calculated using the formula (1). We summarize key properties of T^n in Proposition 1 and Theorem 1. The proof of Proposition 1 is by direct computation and can be found in Appendix A.3.

Proposition 1. *We have the following observations for the block trifocal tensor T^n. For all distinct $i,j \in [n]$, we have the following properties:*

(i) $T^n_{iii} = 0_{3\times3\times3}$

(ii) The T^n_{jii} blocks are rearrangements of elements in the fundamental matrix F_{ij} up to signs.

(iii) The T^n_{iji} and T^n_{iij} blocks encode the epipoles.

(iv) The horizontal slices $T^n(i,:,:)$ of T^n are skew symmetric.

(v) When all cameras are calibrated, three singular values of $T^n_{(1)}$ are equal.

Theorem 1 (Tucker factorization and low multilinear rank of block trifocal tensor)**.** *The block trifocal tensor T^n admits a Tucker factorization, $T^n = \mathcal{G} \times_1 \mathcal{P} \times_2 \mathcal{C} \times_3 \mathcal{C}$, where $\mathcal{G} \in \mathbb{R}^{6\times4\times4}$, $\mathcal{P} \in \mathbb{R}^{3n\times6}$, and $\mathcal{C} \in \mathbb{R}^{3n\times4}$. If the n cameras that produce T^n are not all collinear, then $mlrank(T^n) = (6,4,4)$. If the n cameras that produce T^n are collinear, then $mlrank(T^n) \preceq (6,4,4)$.*

Proof. We can explicitly calculate that $T^n = \mathcal{G} \times_1 \mathcal{P} \times_2 \mathcal{C} \times_3 \mathcal{C}$. The details of the calculation are in Appendix A.2. The specific forms for $\mathcal{G}, \mathcal{C}, \mathcal{P}$ are the following. The horizontal slices of the core are

$$\left\{ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right\}.$$

The factor matrices are $\mathcal{C} = [P_1, P_2, ..., P_n]^T \in \mathbb{R}^{3n\times4}$ and $\mathcal{P} = [S_1, S_2, ..., S_n]^T \in \mathbb{R}^{3n\times6}$, where P_i are the camera matrices and S_i are the corresponding line projection matrices.

Now, we suppose that the n cameras are not collinear. We first show that $\mathcal{C}$ and $\mathcal{P}$ both have full rank. From [1], the null space of a camera matrix P_i is generated by the camera center. For the sake of contradiction, suppose that rank($\mathcal{C}$) < 4. Then there exists $x \in \mathbb{R}^4$ such that $x \neq 0$ and $\mathcal{C}x = 0$. This means that $P_i x = 0$ for all $i = 1,...,n$. Then, x is the camera centre for all cameras, which means that the cameras are centered at one point and are collinear. Similarly, every vector in the null space of the line projection matrix S_i is a line that passes through the camera centre [1]. For the sake of contradiction, suppose that rank($\mathcal{P}$) < 6. Then there exists $x \in \mathbb{R}^6$ such that $x \neq 0$ and $\mathcal{P}x = 0$. This implies that $S_i x = 0$ for all $i = 1,...,n$, which means that x is a line that passes through all of the camera centers. Again the cameras are collinear, which is a contradiction. Next we write the flattening of the block trifocal tensor as $T^n_{(1)} = \mathcal{P}\mathcal{G}_{(1)}(\mathcal{C} \otimes \mathcal{C})^T$. Then $\mathcal{P} \in \mathbb{R}^{3n\times6}$ has rank 6, and $(\mathcal{C} \otimes \mathcal{C})^T \in \mathbb{R}^{16\times9n^2}$ has rank 16. Given the specific form of $\mathcal{G}$, where $\mathcal{G}_{(1)} \in \mathbb{R}^{6\times16}$ it is easy to check rank($\mathcal{G}_{(1)}$) = 6. Thus,

rank($T^n_{(1)}$) = 6. Similarly, we can show that rank($T^n_{(2)}$) = 4, and rank($T^n_{(3)}$) = 4. This implies that the multilinear rank of the block trifocal tensor is $(6,4,4)$ when the n cameras are not collinear.

When the n cameras are collinear, the individual factors in each flattening may be rank deficient, so that rank($T^n_{(1)}$) ≤ 6, rank($T^n_{(2)}$) ≤ 4, and rank($T^n_{(3)}$) ≤ 4. This implies mlrank(T^n) $\preceq (6,4,4)$. □

The theorem inspires a straightforward way of retrieving global poses from the block trifocal tensor, which we summarize in the following claim.

Proposition 2 (One shot camera pose retrieval)**.** *Given the block trifocal tensor T^n produced by cameras $P_1,P_2,...,P_n$, the cameras can be retrieved from T^n up to a global projective ambiguity using the higher-order SVD. The cameras will be the leading 4 singular vectors of $T^n_{(2)}$ or $T^n_{(3)}$.*

Using the higher-order SVD on T^n, we can get a Tucker decomposition of the block trifocal tensor $T^n = \hat{\mathcal{G}} \times_1 \hat{\mathcal{P}} \times_2 \hat{\mathcal{C}} \times_3 \hat{\mathcal{C}}'$. Though the Tucker factorization is not unique [35], as we can apply an invertible linear transformation to one of the factor matrices and apply the inverse onto the core tensor, this invertible linear transformation can be interpreted as the global projective ambiguity for projective 3D reconstruction algorithms. Thus, the cameras can be retrieved by taking the leading four singular vectors of the mode-2 and mode-3 flattenings of the block tensor.

Very importantly however, in practice each trifocal tensor block in T^n can be estimated from image data *only up to an unknown multiplicative scale* [1]. The following theorem establishes the fact that the multilinear rank constraints provide sufficient information for determining the correct scales. In the statement $\odot_b$ denotes blockwise scalar multiplication, thus the (i,j,k)-block of $\lambda \odot_b T^n$ is $\lambda_{ijk} T^n_{ijk} \in \mathbb{R}^{3\times 3\times 3}$.

Theorem 2. *Let $T^n \in \mathbb{R}^{3n\times 3n\times 3n}$ be a block trifocal tensor corresponding to $n \geq 4$ calibrated or uncalibrated cameras in generic position. Let $\lambda \in \mathbb{R}^{n\times n\times n}$ be a block scaling with λ_{ijk} nonzero iff i,j,k are not all equal. Assume that $\lambda \odot_b T^n \in \mathbb{R}^{3n\times 3n\times 3n}$ has multilinear rank $(6,4,4)$ where $\odot_b$ denotes blockwise scalar multiplication. Then there exist $\alpha,\beta,\gamma \in \mathbb{R}^n$ such that $\lambda_{ijk} = \alpha_i\beta_j\gamma_k$ whenever i,j,k are not all the same.*

Sketch. The idea is to identify certain submatrices in the flattenings of $\lambda \odot_b T^n$ which must have determinant 0, and use these to solve for λ. A proof is in Appendix A.4. We remark that the proof technique extends that of [36, Theorem 5.1], which showed a similar result for a matrix problem. □

Theorem 2 is the basic guarantee for our algorithm development below. We stress that the ambiguities brought by α,β,γ are *not* problematic for purposes of recovering the camera matrices by Proposition 2. Indeed, $(\alpha \otimes \beta \otimes \gamma) \odot_b T^n = \mathcal{G} \times_1 (D_\alpha \mathcal{P}) \times_2 (D_\beta \mathcal{C}) \times_3 (D_\gamma \mathcal{C})$ where $D_\alpha \in \mathbb{R}^{3n\times 3n}$ is the diagonal matrix with each entry of α triplicated, etc. Hence the camera matrices can still be recovered up to individual scales (as expected) and a global projective transformation, from the higher-order SVD.

3 Synchronization of the block trifocal tensor

In this section, we develop a heuristic method for synchronizing the block trifocal tensor T^n by exploiting the multilinear rank of T^n from Theorem 1. Let $\hat{T}^n$ denote the estimated block trifocal tensor, and T^n the ground truth. Assume that there are n images and a set of trifocal tensor estimates $\hat{T}_{ijk}$ where $(i,j,k) \in \Omega$ and Ω is the set of indices whose corresponding trifocal tensor is estimated. Note that each estimated trifocal tensor $\hat{T}_{ijk}$ will have an unknown scale $\lambda_{ijk} \in \mathbb{R}^*$ associated with it. We always assume that we observe the iii blocks, as they will be 0. We formulate the block trifocal tensor $\hat{T}^n$ by plugging in the estimates $\hat{T}_{ijk}$ and setting the unobserved positions ($(i,j,k) \notin \Omega$) to $3\times 3\times 3$ tensors of all zeros. Let $W_\Omega \in \{0,1\}^{3n\times 3n\times 3n}$ denote the block tensor where the (i,j,k) blocks are ones for $(i,j,k) \in \Omega$ and zeros otherwise. Let W_{Ω^C} denote the opposite. In our experiments, we observe that the HOSVD is quite robust against noise for retrieving camera poses, which arises e.g., from numerical sensitivities when first estimating relative poses [37]. Therefore we develop an algorithm that projects $\hat{T}^n$ onto the set of tensors that have multilinear rank of $(6,4,4)$ while completing the tensor and retrieving an appropriate set of scales. Specifically, we can write our problem as

$$\min_{\Lambda} \|\Lambda \odot \hat{T}^n - \mathcal{P}_\tau(\Lambda \odot \hat{T}^n)\|^2 \tag{4}$$

where $\Lambda \in \mathbb{R}^{3n\times 3n\times 3n}$, each $3\times 3\times 3$ block is uniform, Λ_{ijk} blocks are zero for $(i,j,k)\notin\Omega$, and Λ satisfies a normalization condition like $\|\Lambda\|^2=1$ to avoid its vanishing. However, we drop this normalization constant in our implementation as we never observe Λ vanishing in practice. (For convenience, we formulate this section with the notation of $\Lambda\in\mathbb{R}^{3n\times 3n\times 3n}$ and Hadamard multiplication, rather than $\lambda\in\mathbb{R}^{n\times n\times n}$ and blockwise scalar multiplication from Theorem 2.) Furthermore in problem (4), $\mathcal{P}_\tau$ denotes the exact projection onto the set $\Gamma=\{T\in\mathbb{R}^{3n\times 3n\times 3n}:\text{mlrank}(T)=(6,4,4)\}$. Note that though HOSVD provides an efficient way to project onto Γ, it is quasi-optimal and not the exact projection. The exact projection is much harder to calculate, and in general NP-hard. The algorithm below adopts an alternating projection strategy to estimate the best set of scales.

3.1 Higher-order SVD with a hard threshold (HOSVD-HT)

The key idea for our algorithm is to use the relative scales on the rank truncated tensor as a heuristic to retrieve scales for the estimated block tensor. There are two main challenges for calculating the rank truncated tensor. First, the exact projection $\mathcal{P}_\tau$ onto Γ is expensive and difficult to calculate. Second, many blocks in the block tensor will be unknown if the corresponding images of the block lacks corresponding point and directly projecting the uncompleted tensor will be inaccurate. We apply an HOSVD framework with imputations to tackle the challenges. Regarding the first challenge, HOSVD is a simple, efficient, and quasi-optimal (3) projection onto Γ. Though inexact, it is a reliable approximation. For the second challenge, the tensor $\hat{T}^n$ must be completed. We adopt the matrix completion idea of HARD-IMPUTE [38], where the matrix is filled-in iteratively with the rank truncated matrix obtained using the hard-thresholded SVD. In other words, we complete the missing blocks with the corresponding blocks in the rank truncated tensor. We define three hyperparameters l_1,l_2,l_3 that correspond to the thresholding parameters of the hard-thresholded SVD on modes 1,2,3 of the block tensor respectively. Specifically, for each mode-i flattening $T^n_{(i)}$, we calculate the full SVD $T^n_{(i)}=USV^T$. Since our tensor will scale cubically with the number of cameras, we suggest using a randomized SVD. We refer to [39] for different randomized strategies. Assume the singular values σ_i on the diagonal of S are sorted in descending order, as usual. We return the factor matrix A_i as the top a left singular vectors in U, where $a=\max\{i:S_{ii}>l_i\}$. Our adapted truncation method is summarized by Algorithm 1.

Algorithm 1 HOSVD-HT

Input: $\hat{T}^n\in\mathbb{R}^{3n\times 3n\times 3n}$: the estimated block tensor; l_1,l_2,$l_3\in\mathbb{R}$: the thresholds for modes 1,2,3 respectively
Output: $\hat{T}_r\in\mathbb{R}^{3n\times 3n\times 3n}$: the rank truncated tensor.
for i = 1 to 3 **do**
  Perform the randomized SVD on the mode-i flattening such that $\hat{T}^n_{(i)}\leftarrow USV^T$
  $a_i\leftarrow\max\{i:S_{ii}>l_i\}$
  $A_i\leftarrow$ first a_i columns of U
end for
$\mathcal{G}=\hat{T}^n\times_1 A_1^T\times_2 A_2^T\times_3 A_3^T$
$\hat{T}^r\leftarrow[\![\mathcal{G};A_1,A_2,A_3]\!]$

From now on, we refer to hard-thresholded HOSVD as HOSVD-HT and denote the operation as $\mathcal{P}_{ht}$.

3.2 Scale recovery

HOSVD-HT provides an efficient way for projecting $\hat{T}^n$ onto the set of tensors with with truncated rank. To recover scales, we use the rank truncated tensor's relative scale as a heuristic to adjust the scale on our estimated block trifocal tensor $\hat{T}^{(n)}$. For each step, we solve

$$\Lambda^{(t+1)}=\underset{\Lambda}{\text{argmin}}\|\Lambda\odot\hat{T}^n-\mathcal{P}_{ht}(\Lambda^{(t)}\odot\hat{T}^n)\|^2\quad\text{s.t. }\Lambda_{ijk}=0_{3\times 3\times 3}\text{ for }(i,j,k)\in\Omega^C, \tag{5}$$

where we drop the normalization condition on Λ because in practice it is not needed. We solve (5) for each observed block separately. Denoting $\mathcal{P}_{ht}(\Lambda^{(t)}\odot\hat{T}^n)$ as $(\hat{T}^n_r)^{(t)}$, we have

$$\Lambda^{(t+1)}_{ijk}=\underset{\mu}{\text{argmin}}\|\mu\cdot\hat{T}^n_{ijk}-(\hat{T}^n_r)^{(t)}_{ijk}\|^2=\frac{trace((\hat{T}^n_{ijk})^T_{(1)}((\hat{T}^n_r)^{(t)}_{ijk})_{(1)})}{\|((\hat{T}^n_r)^{(t)}_{ijk})_{(1)}\|^2_F}. \tag{6}$$

Recall that our strategy for completing the tensor is to impute the tensor with the entries from the rank truncated tensor using HOSVD-HT. Specifically, given the current imputed tensor $(\hat{T}^n)^{(t)}$, we calculate $\mathcal{P}_{ht}((\hat{T}^n)^{(t)})$ and the new scales $\Lambda^{(t+1)}$. Then update with

$$(\hat{T}^n)^{(t+1)} = (\Lambda^{(t+1)} \odot (\hat{T}^n)^{(t)} \odot W_\Omega) + \mathcal{P}_{ht}((\hat{T}^n)^{(t)}) \odot W_{\Omega^C}. \tag{7}$$

3.3 Synchronization algorithm

Now we summarize our synchronization framework in Algorithm 2. We have observed that the

Algorithm 2 Synchronization of the block trifocal tensor

Input: $\hat{T}^n \in \mathbb{R}^{3n\times 3n\times 3n}$; $W_\Omega, W_{\Omega^C} \in \{0,1\}^{3n\times 3n\times 3n}$; $l_1, l_2, l_3 \in \mathbb{R}$
Output: $\mathcal{C} \in \mathbb{R}^{3n\times 4}$: camera matrices up to a 4×4 projective ambiguity and camera-wise scales
Initialize $\hat{T}^n$ by imputing unobserved blocks randomly to get $(\hat{T}^n)^{(0)}$
while not converged **do**
  Calculate $\mathcal{P}_{ht}((\hat{T}^n)^{(t)})$ using HOSVD-HT
  Calculate $\Lambda^{(t+1)}$ (5) using (6)
  $(\hat{T}^n)^{(t+1)} \leftarrow (\Lambda^{(t+1)} \odot (\hat{T}^n)^{(t)} \odot W_\Omega) + \mathcal{P}_{ht}((\hat{T}^n)^{(t)}) \odot W_{\Omega^C}$
  $t \leftarrow t+1$
end while
$(\mathcal{G}, A_1, A_2, A_3) \leftarrow HOSVD((\hat{T}^n)^{(t)})$
$\mathcal{C} \leftarrow$ First 4 columns of A_2

algorithm can overfit, as the recovered scales will experience sudden and huge leaps. Our stopping criteria for the algorithm is when we observe sudden jumps in the variance of the new scales or when we exceed a maximum number of iterations. Another challenge in structure from motion datasets is that estimations may be highly corrupted. The HOSVD framework mainly consists of retrieving a dominant subspace from each flattening. Thus, it is natural to replace the SVD on each flattening with a more robust subspace recovery method, such as the Tyler's M estimator (TME) [40] or a recent extension of TME that incorporates the information of the dimension of the subspace in the algorithm [41]. We refer to Appendix A.5.2 for more details and provide an implementation there.

4 Numerical experiments

We conduct experiments of Algorithm 2 on two benchmark real datasets, the EPFL datasets [42] and the Photo Tourism datasets [11]. We observe that the algorithm performs better in the calibrated setting, and since the calibration matrix is usually known in practice, we restrict our scope of experiments to calibrated trifocal tensors. We compare against three state-of-the-art synchronization based on two view measurements, NRFM [18] and LUD [12]. NRFM relies on nonconvex optimization and requires a good initialization. We test NRFM with an initialization obtained from LUD and with a random initialization. We also test BATA [43] initialized with MPLS [9]. We refer to A.6 in the appendix for a comprehensive summary of numerical results including rotation and translation estimation errors. We include our code in the following github repository: TrifocalSync.

4.1 EPFL dataset

For EPFL, we follow the experimental setup and adopt code from [44] and test an entire structure from motion pipeline. We first describe the structure from motion pipeline for EPFL experiements.

• Step 1 (feature detection and feature matching). We obtain matched features across pairs of images using a modern deep learning based feature detection and matching algorithm, GlueStick [45]. Though we do not implement this in our experiments, there have been methods developed to further screen corrupted keypoint matches or obtain matches robustly, such as [46, 47, 48]. Key points across a triplet of cameras is matched from pairs and is included only if it appears in all the pair combinations of the three images.

• Step 2 (estimation and refinement of trifocal tensors). With the triplet matches, we calculate the trifocal tensors with more than 11 correspondences. To have an even sparser graph, one can skip the

estimation of trifocal tensors and rely on the imputation for images that have less than a number bigger than 11 point correspondences. This can further speed up the trifocal tensor estimation process. We apply STE from [41] to find 40% of the correspondences as inliers, then use at most 30 inlier point correspondences to linearly estimate the trifocal tensor. To refine the estimates, we apply bundle adjustment on the inliers and delete triplets with reprojection error larger than 1 pixel.

• Step 3 (synchronization). We synchronize the estimated block trifocal tensor with a robust variant of SVD using the framework described in Algorithm 2. The robustness comes from replacing SVD with a robust subspace recovery method [41]. More details can be found in Appendix A.5.2. Recall that the cameras we retrieve are up to a global projective ambiguity. When comparing with ground truth poses, we first align our estimated cameras with the ground truth cameras by finding a 4×4 projective transformation. Then we round the cameras to calibrated cameras and compare.

We test our full pipeline on two EPFL datasets on a personal machine with 2 GHz Intel Core i5 with 4 cores and 16GB of memory. To test NRFM [18], LUD [12] and BATA [43] initialized with MPLS [9], we estimate the corresponding essential matrices using the GC-ransac [49]. We did not include blocks corresponding to two views in our trifocal tensor pipeline. The mean and median translation errors are summarized in Figure 1 here and more comprehensive results can be found in Table 1 and Table 2 in the appendix.

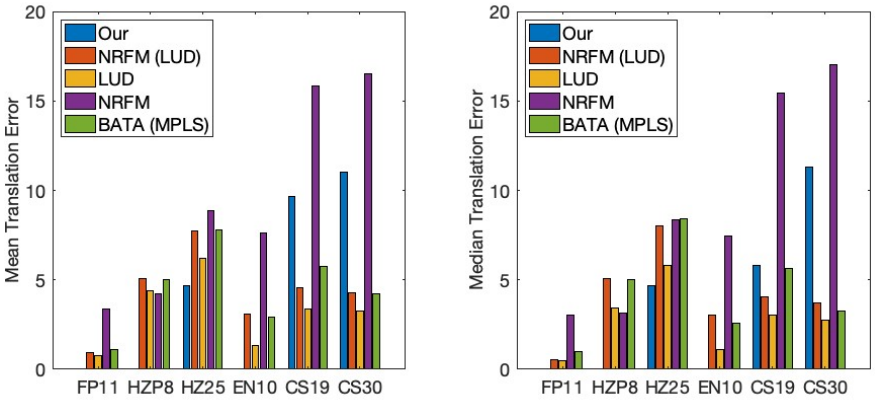


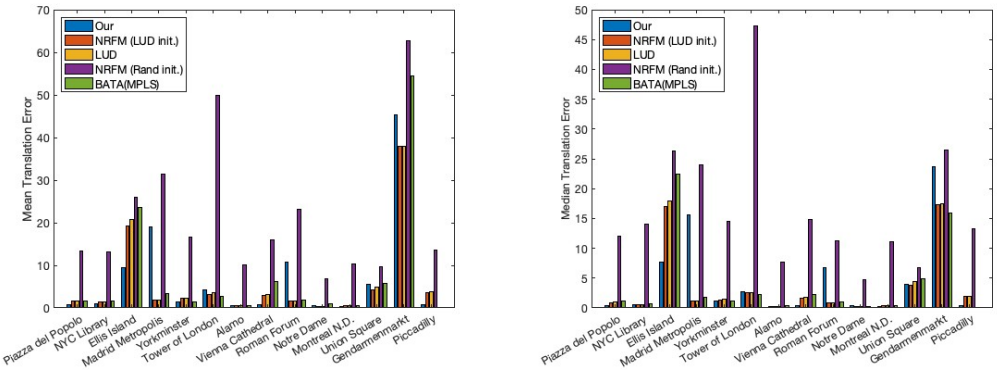
Figure 1: EPFL translation error comparison between our method, NRFM initialized by LUD, LUD, and NRFM initialized randomly. BATA(MPLS) stands for BATA initialized by MPLS. HZ8 stands for HerzP8, FP11 for FountainP11, HZ25 for Herz P25, EN10 for EntryP10, CS19 for CastleP19, CS30 for CastleP30.

The EPFL datasets generally have a plethora of point correspondences, so that the trifocal tensors are estimated accurately. When the dataset focuses on a single scene, our algorithm retrieves locations competitively. Our algorithm achieves the best location estimation for 4 out of 6 datasets. The translation error bars are not visible for FP11, HZP8, EN10 due to the accuracy that we achieve. However, our pipeline is incapable of accurately processing CastleP19 and CastleP30. The main reason is that our algorithm relies on having a very dense observation graph to ensure high completion rate. CastleP19 and CastleP30 are datasets where the camera scans portions of the general area sequentially, so that not many triplets have overlapping features. Our method is not suitable for this type of dataset. However, it is possible to apply our algorithm in parallel on groups of neighboring frames, so that the completion rate is high in each group. Then the results can be merged to obtain a larger reconstruction. Rotations for the two view methods are estimated via rejecting outliers from iteratively applying [10]. We also compare against [43] for location estimation, where we initialize with a state-of-the-art global rotation estimation method [9]. Our algorithm achieves superior rotation estimation for only 2 out of the 6 datasets. See Table 1 and 2 in the appendix for comprehensive errors.

4.2 Photo Tourism

We conduct experiments on the Photo Tourism datasets. The Photo Tourism datasets consist of internet images of real world scenes. Each scene has hundreds to thousands of images. The datasets

[11] provide essential matrix estimates, and we estimate the trifocal tensors from the given essential matrices. To limit the computational cost for tensors, we downsample the datasets by choosing cameras with observations more than a certain percentage in the corresponding block frontal slice while maintaining a decent number of cameras. Note that this may not be the optimal way of extracting a dense subset in general. The maximum number of cameras we select for each dataset is 225 cameras. The largest dataset Piccadilly has 2031 cameras initially. We randomly sample 1000 cameras and then run our procedure. For Roman Forum and Piccadilly, the two view methods further deleted cameras from the robust rotation estimation process or parallel rigidity test. We rerun and report the trifocal tensor synchronization algorithm with the further downsampled data. We initialize the hard thresholding parameters for HOSVD-HT by first imputing the trifocal tensor with small random entries and then calculating the singular values for each of the flattenings. We take l_i to be the tertile singular value for each mode-i flattening. We then keep this parameter fixed for the synchronization process. Recall that the jii blocks in the block trifocal tensor correspond to elements in the essential matrix E_{ij}. We also include these essential matrix estimations in the block trifocal tensor. The Photo Tourism experiments were run on an HPC center with 32 cores, but the only procedure that can benefit from parallel computing in a single experiment is the scale retrieval. Mean and median translation errors are summarized in Figure 2. Fully comprehensive results can be found in Tables 3 and 4 in Appendix A.6.



(a) Photo Tourism mean translation errors

(b) Photo Tourism median translation errors

Figure 2: Photo Tourism translation error comparison between our method, NRFM initialized by LUD, LUD, NRFM initialized randomly, and BATA initialized with MPLS. Note that we have not been able to acquire results for Piccadilly for BATA + MPLS.

Our method is able to achieve competitive translation errors on 8 of the 14 datasets tested. Similar to the observation in the EPFL experiments, our algorithm performs well when the viewing graph is dense, or in other words, when the estimation percentage is high. We achieve better locations in 6 out of 8 datasets where the estimation percentage exceeds 60%, and better locations in only 2 out of 6 datasets where the estimation percentage falls below 60%. We achieve reasonable rotation estimations for 10 out of 14 datasets, but not as good as LUD. See Table 4 for a comprehensive result. Since the block trifocal tensor scales cubically with respect to the number of cameras, our algorithm runtime is longer than most two view global methods. This could be alleviated by synchronizing dense subsets in parallel and merging the results to construct a larger reconstruction.

Additional remark: Trifocal tensors can be estimated from line correspondences or a mix of point and line correspondences, while fundamental matrices are estimated from only point correspondences. There are many situations where accurate point correspondences are in short supply but there is a plethora of clear and distinct lines. For example, see datasets in a recent SfM method using lines [50]. We demonstrate the potential of our method to be adapted to process datasets with only lines or very few points. Due to the limited availability of well annotated line datasets, we provide a small synthetic experiment that simulates a case where only lines correspondences are present. We first generate 20 random camera matrices, then we generate 25 lines that are projected on and shared across all images. We add about 0.02 percent of noise in terms of the relative frobenius norms between the line equation parameters and the noise. We estimate the trifocal tensor of three different views from line correspondences linearly. One remark is that our synchronization method works well only when the signs of the initial unknown scales are mostly uniform. We manually use ground truth trifocal tensors

to correct the sign of the scale. This has not been an issue in the previous experiments due to bundle adjustment for EPFL and the overall good estimations in Photo Tourism. In practice, the sign of the scale on a trifocal tensor can be corrected via triangulation of points or reconstruction of lines, and correcting the sign using the depths of the reconstructed points or intersecting line segments. We synchronize the trifocal tensors with Algorithm 2 and were able to achieve a mean rotation error of 0.61 degrees, median rotation error of 0.49 degrees, mean location error of 0.76, and median location error of 0.74.

5 Conclusion

In this work, we introduced the block tensor of trifocal tensors characterizing the three-view geometry of a scene. We established an explicit Tucker factorization of the block trifocal tensor and proved it has a low multilinear rank of $(6,4,4)$ under appropriate scaling. We developed a synchronization algorithm based on tensor decomposition that retrieves an appropriate set of scales, and synchronizes rotations and translations simultaneously. On several real data benchmarks we demonstrated state-of-the-art performance in terms of camera location estimation, and saw particular advantages on smaller and denser sets of images. Overall, this work suggests that higher-order interactions in synchronization problems have the potential to improve performance over pairwise-based methods.

There are several limitations to our tensor-based synchronization method. First, our rotation estimations are not as strong as our location estimations. Second, our algorithm performance is affected by the estimation percentage of trifocal tensors within the block trifocal tensor. One could incorporate more robust completion methods and explore new approaches for processing sparse triplet graphs. Further, our block trifocal tensor scales cubically in terms of the number of cameras and becomes computationally expensive for large datasets. We can develop methods for extracting dense subgraphs, synchronizing in parallel, then merging results to obtain a larger reconstruction, similarly to the distributed algorithms of [51] and [52]. Moreover, our synchronization method's success depends on accurate trifocal tensor estimations, and it motivates further work on robust estimation of multi-view tensors. Algorithm 2 could also be made more robust by adding outlier rejection techniques. Finally we plan to extend our theory by proving convergence of our algorithm and exploring structures for even higher-order tensors, such as quadrifocal tensors.

Acknowledgement

D.M. and G.L. were supported in part by NSF award DMS 2152766. J.K. was supported in part by NSF awards DMS 2309782 and CISE-IIS 2312746, the DOE award SC0025312, and start-up grants from the College of Natural Science and Oden Institute at the University of Texas at Austin.

We thank Shaohan Li and Feng Yu for helpful discussions on processing EPFL and Photo Tourism. We also thank Hongyi Fan for helpful advice and references on estimating trifocal tensors.

References

[1] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[2] Noah Snavely, Steven Seitz, and Richard Szeliski. Photo Tourism: Exploring photo collections in 3D. In *Proceedings of the ACM Special Interest Group on Computer Graphics and Interactive Techniques Conference, SIGGRAPH 2006*, pages 835–846, 2006.

[3] Johannes Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 4104–4113, 2016.

[4] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment — A modern synthesis. In Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice*, pages 298–372, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[5] Venu Madhav Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2004*, volume 1, pages 1–8, 2004.

[6] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International Journal of Computer Vision*, 103:267–305, 2013.

[7] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):958–972, 2018.

[8] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV 2013*, pages 521–528, 2013.

[9] Yunpeng Shi and Gilad Lerman. Message passing least squares framework and its application to rotation synchronization. In *Proceedings of the International Conference on Machine Learning, ICML 2020*, pages 8796–8806, 2020.

[10] Mica Arie-Nachimson, Shahar Kovalsky, Ira Kemelmacher-Shlizerman, Amit Singer, and Ronen Basri. Global motion estimation from point matches. In *Proceedings of the International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, 3DIMPVT 2012*, pages 81–88, 2012.

[11] Kyle Wilson and Noah Snavely. Robust global translations with 1DSfM. In *Proceedings of the European Conference on Computer Vision, EECV 2014*, pages 61–75, 2014.

[12] Onur Ozyesil and Amit Singer. Robust camera location estimation by convex programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pages 2674–2683, 2015.

[13] Thomas Goldstein, Paul Hand, Choongbum Lee, Vladislav Voroninski, and Stefano Soatto. ShapeFit and ShapeKick for robust, scalable structure from motion. In *Proceedings of the European Conference on Computer Vision, EECV 2016*, pages 289–304, 2016.

[14] David Rosen, Luca Carlone, Afonso Bandeira, and John Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group. *International Journal of Robotics Research*, 38(2-3):95–125, 2019.

[15] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Spectral synchronization of multiple views in SE(3). *SIAM Journal on Imaging Sciences*, 9(4):1963–1990, 2016.

[16] Mihai Cucuringu, Yaron Lipman, and Amit Singer. Sensor network localization by eigenvector synchronization over the Euclidean group. *ACM Transactions on Sensor Networks*, 8(3), 2012.

[17] Jesus Briales and Javier Gonzalez-Jimenez. Cartan-Sync: Fast and global SE(d)-synchronization. *IEEE Robotics and Automation Letters*, 2(4):2127–2134, 2017.

[18] Soumyadip Sengupta, Tal Amir, Meirav Galun, Tom Goldstein, David Jacobs, Amit Singer, and Ronen Basri. A new rank constraint on multi-view fundamental matrices, and its application to camera location recovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pages 4798–4806, 2017.

[19] Yoni Kasten, Amnon Geifman, Meirav Galun, and Ronen Basri. Algebraic characterization of essential matrices and their averaging in multiview settings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 5895–5903, 2019.

[20] Yoni Kasten, Amnon Geifman, Meirav Galun, and Ronen Basri. GPSfM: Global projective SFM using algebraic constraints on multi-view fundamental matrices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 3264–3272, 2019.

[21] Spyridon Leonardos, Roberto Tron, and Kostas Daniilidis. A metric parametrization for trifocal tensors with non-colinear pinholes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pages 259–267, 2015.

[22] Viktor Larsson, Nicolas Zobernig, Kasim Taskin, and Marc Pollefeys. Calibration-free structure-from-motion with calibrated radial trifocal tensors. In *Proceedings of the European Conference on Computer Vision, EECV 2020*, pages 382–399, 2020.

[23] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure-from-motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pages 3248–3255, 2013.

[24] Onur Ozyesil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017.

[25] Joe Kileel. Minimal problems for the calibrated trifocal variety. *SIAM Journal on Applied Algebra and Geometry*, 1(1):575–598, 2017.

[26] Timothy Duff, Kathlen Kohn, Anton Leykin, and Tomas Pajdla. PLMP-point-line minimal problems in complete multi-view visibility. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 1675–1684, 2019.

[27] Ricardo Fabbri, Timothy Duff, Hongyi Fan, Margaret Regan, David da Costa de Pinho, Elias Tsigaridas, Charles Wampler, Jonathan Hauenstein, Peter Giblin, Benjamin Kimia, Anton Leykin, and Tomas Pajdla. Trifocal relative pose from lines at points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7870–7884, 2023.

[28] David Nister and Henrik Stewenius. A minimal solution to the generalised 3-point pose problem. *Journal of Mathematical Imaging and Vision*, 27(1):67–79, 2007.

[29] Ali Elqursh and Ahmed Elgammal. Line-based relative pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011*, pages 3049–3056, 2011.

[30] Yubin Kuang and Kalle Astrom. Pose estimation with unknown focal length using points, directions and lines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pages 529–536, 2013.

[31] Zuzana Kukelova, Joe Kileel, Bernd Sturmfels, and Tomas Pajdla. A clever elimination strategy for efficient minimal solvers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pages 4912–4921, 2017.

[32] Pedro Miraldo, Tiago Dias, and Srikumar Ramalingam. A minimal closed-form solution for multi-perspective pose estimation using points and lines. In *Proceedings of the European Conference on Computer Vision, ECCV 2018*, pages 474–490, 2018.

[33] Joe Kileel, Zuzana Kukelova, Tomas Pajdla, and Bernd Sturmfels. Distortion varieties. *Foundations of Computational Mathematics*, 18:1043–1071, 2018.

[34] Joe Kileel and Kathlén Kohn. Snapshot of algebraic vision. *arXiv preprint arXiv:2210.11443*, 2022.

[35] Tamara Kolda and Brett Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[36] Tommi Muller, Adriana Duncan, Eric Verbeke, and Joe Kileel. Algebraic constraints and algorithms for common lines in cryo-EM. *Biological Imaging*, pages 1–30, Published online 2024.

[37] Hongyi Fan, Joe Kileel, and Benjamin Kimia. On the instability of relative pose estimation and RANSAC's role. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2022*, pages 8935–8943, 2022.

[38] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010.

[39] Nathan Halko, Per-Gunnar Martinsson, and Joel Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[40] David Tyler. A distribution-free M-estimator of multivariate scatter. *Annals of Statistics*, pages 234–251, 1987.

[41] Feng Yu, Teng Zhang, and Gilad Lerman. A subspace-constrained Tyler's estimator and its applications to structure from motion. *arXiv preprint arXiv:2404.11590*, 2024.

[42] Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008*, pages 1–8, 2008.

[43] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Baseline desensitizing in translation averaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018)*, June 2018.

[44] Laura Julia and Pascal Monasse. A critical review of the trifocal tensor estimation. In *Proceedings of the Pacific Rim Symposium on Image and Video Technology, PSIVT 2017, Revised Selected Papers 8*, pages 337–349. Springer, 2018.

[45] Rémi Pautrat, Iago Suárez, Yifan Yu, Marc Pollefeys, and Viktor Larsson. Gluestick: Robust image matching by sticking points and lines together. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, CVPR 2023*, pages 9706–9716, 2023.

[46] Yunpeng Shi, Shaohan Li, Tyler Maunu, and Gilad Lerman. Scalable cluster-consistency statistics for robust multi-object matching. In *Proceedings of the International Conference on 3D Vision, 3DV 2021*, pages 352–360, 2021.

[47] Yunpeng Shi, Shaohan Li, and Gilad Lerman. Robust multi-object matching via iterative reweighting of the graph connection laplacian. *Advances in Neural Information Processing Systems*, 33:15243–15253, 2020.

[48] Shaohan Li, Yunpeng Shi, and Gilad Lerman. Fast, accurate and memory-efficient partial permutation synchronization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2022*, pages 15735–15743, 2022.

[49] Daniel Barath and Jiří Matas. Graph-cut ransac. In *Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR 2018*, pages 6733–6741, 2018.

[50] Shaohui Liu, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. 3d line mapping revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023*, pages 21445–21455, 2023.

[51] Shaohan Li, Yunpeng Shi, and Gilad Lerman. Efficient detection of long consistent cycles and its application to distributed synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024*, pages 5260–5269, 2024.

[52] Andrea Porfiri Dal Cin, Luca Magri, Federica Arrigoni, Andrea Fusiello, and Giacomo Boracchi. Synchronization of group-labelled multi-graphs. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021*, pages 6433–6443. IEEE, 2021.

[53] Joe Harris. *Algebraic Geometry: A First Course*, volume 133. Springer Science & Business Media, 1992.

[54] Ying Sun, Prabhu Babu, and Daniel Palomar. Regularized Tyler's scatter estimator: Existence, uniqueness, and algorithms. *IEEE Transactions on Signal Processing*, 62(19):5143–5156, 2014.

A Appendix / supplemental material

A.1 Derivation of the trifocal tensor

To provide a better intuition for the trifocal tensor, we briefly summarize the derivation of the trifocal tensor from [1] and [21] under the general setup of uncalibrated cameras.

Let $P_i = K_i R_i [I, -t_i]$ be the form of the camera matrix for P_1, P_2, P_3. Let L be a line in the 3D world scene, and l_1, l_2, l_3 the corresponding projections in the images I_1, I_2, I_3 respectively. Each l_i back projects to a plane $\pi_i = P_i^T l_i$ in $\mathbb{R}^3$, and since l_i correspond to the same L in the 3D world scene, $\pi = [\pi_1, \pi_2, \pi_3]$ must be rank deficient and its kernel will generically be spanned by L. Then,

$$\pi' = \begin{bmatrix} K_1^{-T} R_1 & 0 \\ t_1^T & 1 \end{bmatrix} \pi = \begin{bmatrix} l_1 & K_1^{-T} R_{12} K_2^T l_2 & K_1^{-T} R_{13} K_3^T l_3 \\ 0 & (t_1 - t_2)^T R_2^T K_2^T l_2 & (t_1 - t_3)^T R_3^T K_3^T l_3 \end{bmatrix} = [\pi'_1, \pi'_2, \pi'_3]$$

will also be rank-deficient, implying that the columns of π' are linearly dependent. This means that there exist α, β such that $\pi'_1 = \alpha \pi'_2 + \beta \pi'_3$. We can choose $\alpha = -(t_1 - t_3)^T R_3^T K_3^T l_3$, $\beta = (t_1 - t_2)^T R_2^T K_2^T l_2$, so that

$$l_1 = l_2^T [K_2 R_2 (t_1 - t_2) K_1^{-T} R_{13} K_3^T] l_3 - l_2^T [K_2 R_{12}^T K_1^{-1} (t_1 - t_3)^T R_3^T K_3^T] l_3.$$

Then, the canonical trifocal tensor centered at camera 1 is defined as

$$T_i = K_2 R_2 (t_1 - t_2) e_i^T K_1^{-T} R_{13} K_3^T - K_2 R_{12}^T K_1^{-1} e_i (t_1 - t_3)^T R_3^T K_3^T \tag{8}$$

where $e_i \in \mathbb{R}^3$ is the i-th standard basis vector. The trifocal tensor will be the tensor $\{T_1, T_2, T_3\}$, where the T_i's are stacked along the first mode. The line incidence relation is then $(l_1)_i = l_2^T T_i l_3$. Other combinations of point and line incidence relations are also encoded by the trifocal tensor; see [1] for details. The construction for calibrated cameras is the same, just with P_i in calibrated form.

A.2 Proof details for Theorem 1

We include a detailed calculation for the Tucker factorization of the block trifocal tensor. Recall that each individual trifocal tensor corresponding to the cameras a, b, c can be calculated as

$$\begin{aligned}
T_{iqr} &= (-1)^{i+1} \det \begin{bmatrix} \sim a^i \\ b^q \\ c^r \end{bmatrix} = (-1)^{i+1} \det \begin{bmatrix} a^m \\ a^n \\ b^q \\ c^r \end{bmatrix} \\
&= (-1)^{i+1} (\det \begin{bmatrix} a_{m3} & a_{m4} \\ a_{n3} & a_{n4} \end{bmatrix} (b_{q1} c_{r2} - b_{q2} c_{r1}) + \det \begin{bmatrix} a_{m2} & a_{m4} \\ a_{n2} & a_{n4} \end{bmatrix} (-b_{q1} c_{r3} + b_{q3} c_{r1}) \\
&\quad + \det \begin{bmatrix} a_{m1} & a_{m4} \\ a_{n1} & a_{n4} \end{bmatrix} (b_{q2} c_{r3} - b_{q3} c_{r2}) + \det \begin{bmatrix} a_{m2} & a_{m3} \\ a_{n2} & a_{n3} \end{bmatrix} (b_{q1} c_{r4} - b_{q4} c_{r1}) \\
&\quad + \det \begin{bmatrix} a_{m1} & a_{m3} \\ a_{n1} & a_{n3} \end{bmatrix} (-b_{q2} c_{r4} + b_{q4} c_{r2}) + \det \begin{bmatrix} a_{m1} & a_{m2} \\ a_{n1} & a_{n2} \end{bmatrix} (b_{q3} c_{r4} - b_{q4} c_{r3})) \\
&= \mathcal{P}(a)_{i6} (b_{q1} c_{r2} - b_{q2} c_{r1}) + \mathcal{P}(a)_{i5} (-b_{q1} c_{r3} + b_{q3} c_{r1}) + \mathcal{P}(a)_{i3} (b_{q2} c_{r3} - b_{q3} c_{r2}) \\
&\quad + \mathcal{P}(a)_{i4} (b_{q1} c_{r4} - b_{q4} c_{r1}) + \mathcal{P}(a)_{i2} (-b_{q2} c_{r4} + b_{q4} c_{r2}) + \mathcal{P}(a)_{i1} (b_{q3} c_{r4} - b_{q4} c_{r3}) \\
&= \sum_{k=1}^{6} \mathcal{P}(a)_{ik} \sum_{w=1}^{4} b_{qw} \sum_{j=1}^{4} c_{rj} \mathcal{G}_{kwj}.
\end{aligned}$$

The last equality can be easily checked since $\mathcal{G}$ is sparse. For example, when $k = 1$, $\mathcal{P}(a)_{i1}$ is the determinant of the submatrix dropping the i-th row and keeping columns 1 and 2, which is $\det[a_{m1} a_{m2}; a_{n1} a_{n2}]$. The only nonzero elements in the first horizontal slice are $\mathcal{G}(1, 4, 3) = -1$ and $\mathcal{G}(1, 3, 4) = 1$. Then, the nonzero elements in the sum when $k = 1$ will be exactly $\mathcal{P}(a)_{i1} \sum_{w=1}^{4} b_{qw} \sum_{j=1}^{4} c_{rj} \mathcal{G}_{1wj} = \mathcal{P}(a)_{i1} (b_{q3} c_{r4} - b_{q4} c_{r3})$.

Then, since $\mathcal{P}$ will be the stackings of $\mathcal{P}(P_i)$, $\mathcal{C}$ is the stacking of camera matrices in Theorem 1, each ijk block in T^n will be calculated by exactly the corresponding i, j, k blocks in $\mathcal{P}, \mathcal{C}, \mathcal{C}$ respectively using the calculations above.

A.3 Proof details for Proposition 1

Proof for (i). We have

$$(T^n_{iii})_{wqr} = (-1)^{w+1}\det\begin{bmatrix}\sim P^w_i \\ P^q_i \\ P^r_i\end{bmatrix} = 0,$$

since P_i is a 3×4 matrix and the submatrix above will always have two identical rows.

Proof for (ii): Consider the wqr element of the jii block trifocal tensor, T^n_{jii}. It can be written as

$$(T^n_{jii})_{wqr} = (-1)^{w+1}\det\begin{bmatrix}\sim P^w_j \\ P^q_i \\ P^r_i\end{bmatrix}.$$

Thus, when $q = r$, clearly $(T^n_{jii})_{wqr} = 0$ as we will have identical rows again. When $q \neq r$, we first observe that $(T^n_{jii})_{wqr} = -(T^n_{jii})_{wrq}$ since we just swap two rows. Second,

$$(T^n_{jii})_{wqr} = (-1)^{w+1}\det\begin{bmatrix}\sim P^w_j \\ P^q_i \\ P^r_i\end{bmatrix} = (-1)^{w+1}\det\begin{bmatrix}\sim P^w_j \\ \sim P^m_i\end{bmatrix}$$

where $m \in \{1,2,3\}\setminus\{q,r\}$. This is exactly the bilinear relationship in [1] defining the fundamental matrix $(F_{ji})_{mw}$ element up to a possible negative sign.

Proof for (iii): We can only show this for T^n_{iij} blocks from symmetry. The elements in T^n_{iij} blocks can be calculated as

$$(T^n_{iji})_{wqr} = (-1)^{w+1}\det\begin{bmatrix}\sim P^w_i \\ P^q_i \\ P^r_j\end{bmatrix}$$

Elements are nonzero only when $w = q$, and they correspond to determinants of matrices with three rows from one P_i and one row from P_j. By [1], these are exactly the elements of the epipoles. When $w = 1$, the order of the rows in the determinant corresponding to camera i is $(2,3,1)$, when $w = 2$, the order is $(1,3,2)$ and there is a negative sign in front of the determinant, and when $w = 3$, the order is $(1,2,3)$. Since the first and last case are even permutations of the rows of P_i, and the second case is corrected by a negative sign, $(T^n_{iji})_{ww:}$ is exactly the epipole.

Proof for (iv): On a horizontal slice, the camera along the 1st mode is fixed, and blocks symmetric across the diagonal is calculated by cameras, which the 2nd and 3rd mode cameras are swapped. Then, we will simply be swapping rows in (1), which means that we will simply be changing signs for elements symmetric across the diagonal, implying skew symmetry.

Proof for (v): Now assume that we have a block trifocal tensor whose corresponding cameras are all calibrated. Let $\mathcal{P}$ be the line projection matrix, $\mathcal{C} = [P_1, P_2, \dots, P_n]^T$ is the stacked camera matrix, and $\mathcal{G}$ is the core tensor. The flattening in the 1st mode can be written as $T^n_{(1)} = \mathcal{P}\mathcal{G}_{(1)}(\mathcal{C}\otimes\mathcal{C})^T$, where $T^n_{(1)}$ is a $3n \times 9n^2$ matrix. For the proof, we calculate the eigenvalue of $T^n_{(1)}(T^n_{(1)})^T = \mathcal{P}\mathcal{G}_{(1)}(\mathcal{C}\otimes\mathcal{C})^T(\mathcal{C}\otimes\mathcal{C})\mathcal{G}^T_{(1)}\mathcal{P}^T$

$$\begin{aligned} T^n_{(1)}(T^n_{(1)})^T &= \mathcal{P}\mathcal{G}_{(1)}(\mathcal{C}\otimes\mathcal{C})^T(\mathcal{C}\otimes\mathcal{C})\mathcal{G}^T_{(1)}\mathcal{P}^T \\ &= \mathcal{P}\mathcal{G}_{(1)}(\mathcal{C}^T\otimes\mathcal{C}^T)(\mathcal{C}\otimes\mathcal{C})\mathcal{G}^T_{(1)}\mathcal{P}^T \\ &= \mathcal{P}\mathcal{G}_{(1)}(\mathcal{C}^T\mathcal{C}\otimes\mathcal{C}^T\mathcal{C})\mathcal{G}^T_{(1)}\mathcal{P}^T. \end{aligned}$$

The second and third line uses two Kronecker product properties: $(A\otimes B)^T = A^T\otimes B^T$ and $(A\otimes B)(C\otimes D) = AC\otimes BD$ as long as AC and BD are defined.

We first calculate $(\mathcal{C}^T\mathcal{C}\otimes\mathcal{C}^T\mathcal{C})$.

We assume that the cameras are centered at the origin, i.e. $\sum_{i=1}^n t_i = 0$. Then we have

$$\mathcal{C}^T\mathcal{C} = \begin{bmatrix} nI_{3\times 3} & -\sum_{i=1}^n t_i \\ -\sum_{i=1}^n t_i^T & \sum_{i=1}^n \|t_i\|^2 \end{bmatrix} = \begin{bmatrix} nI_{3\times 3} & 0_{3\times 1} \\ 0_{1\times 3} & \sum_{i=1}^n \|t_i\|^2 \end{bmatrix}, \tag{9}$$

so that

$$(\mathcal{C}^T\mathcal{C}\otimes\mathcal{C}^T\mathcal{C})=\begin{bmatrix} nI_{3\times 3}\otimes\mathcal{C}^T\mathcal{C} & 0_{3\times 1}\otimes\mathcal{C}^T\mathcal{C} \\ 0_{1\times 3}\otimes\mathcal{C}^T\mathcal{C} & \sum_{i=1}^n\|t_i\|^2\otimes\mathcal{C}^T\mathcal{C}\end{bmatrix} \tag{10}$$

We have an explicit form for $\mathcal{G}_{(1)}$:

$$\mathcal{G}_{(1)}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{11}$$

Let $X=\mathcal{C}^T\mathcal{C}$ and let X_{ij} denote the ij entry in $\mathcal{C}^T\mathcal{C}$. Let $a=\sum_{i=1}^n\|t_i\|^2$.

We first show that $\mathcal{G}_{(1)}(\mathcal{C}^T\mathcal{C}\otimes\mathcal{C}^T\mathcal{C})\mathcal{G}_{(1)}^T$ is diagonal by direct computation:

$$\begin{aligned}
&\mathcal{G}_{(1)}(\mathcal{C}^T\mathcal{C}\otimes\mathcal{C}^T\mathcal{C})\mathcal{G}_{(1)}^T \\
&=\begin{bmatrix} nX_{44}+aX_{33} & aX_{22} & -nX_{42} & aX_{31} & nX_{41} & 0 \\ -aX_{23} & nX_{44}+aX_{22} & nX_{43} & aX_{21} & 0 & nX_{41} \\ nX_{24} & -nX_{34} & nX_{33}+nX_{22} & 0 & -nX_{21} & -nX_{31} \\ aX_{13} & -aX_{12} & 0 & nX_{44}+aX_{11} & -nX_{43} & nX_{42} \\ nX_{14} & 0 & -nX_{12} & nX_{34} & nX_{33}+nX_{11} & -nX_{32} \\ 0 & -nX_{24} & -nX_{13} & nX_{24} & -nX_{23} & nX_{22}+nX_{11} \end{bmatrix} \\
&=\begin{bmatrix} na+an & 0 & 0 & 0 & 0 & 0 \\ 0 & na+an & 0 & 0 & 0 & 0 \\ 0 & 0 & n^2+n^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & na+an & 0 & 0 \\ 0 & 0 & 0 & 0 & n^2+n^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & n^2+n^2 \end{bmatrix} \\
&=\begin{bmatrix} 2na & 0 & 0 & 0 & 0 & 0 \\ 0 & 2na & 0 & 0 & 0 & 0 \\ 0 & 0 & 2n^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2na & 0 & 0 \\ 0 & 0 & 0 & 0 & 2n^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2n^2 \end{bmatrix}.
\end{aligned}$$

We then calculate the spectral decomposition of $T_{(1)}^n$. With a slight abuse of notation, let P_i^k denote the k-th column of P_i. The $3n\times 6$ rank-6 stacked line projection matrix would have columns ordered according to

$$[e_1\wedge e_2 \quad e_1\wedge e_3 \quad e_1\wedge e_4 \quad e_2\wedge e_3 \quad e_2\wedge e_4 \quad e_3\wedge e_4],$$

and since the second row in $\mathcal{P}$ for each camera is $P_i^3\wedge P_i^1$ it holds

$$\mathcal{P}=\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_i^1\times P_i^2 & P_i^1\times P_i^3 & P_i^1\times P_i^4 & P_i^2\times P_i^3 & P_i^2\times P_i^4 & P_i^3\times P_i^4 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}.$$

Or equivalently, the stacked wedge products between columns. Let $\mathcal{P}=USV^T$ be the thin singular value decomposition of $\mathcal{P}$, so that U is a $3n\times 6$ orthonormal matrix, S is a 6×6 diagonal matrix where all diagonal entries are nonzero, and V is a 6×6 orthonormal matrix.

Then,

$$\begin{aligned}
T_{(1)}^n(T_{(1)}^n)^T&=\mathcal{P}\mathcal{G}_{(1)}(\mathcal{C}^T\mathcal{C}\otimes\mathcal{C}^T\mathcal{C})\mathcal{G}_{(1)}^T\mathcal{P}^T \\
&=U(SV^T\mathcal{G}_{(1)}(\mathcal{C}^T\mathcal{C}\otimes\mathcal{C}^T\mathcal{C})\mathcal{G}_{(1)}^TVS)U^T.
\end{aligned}$$

Since V is orthonormal, $SV^T\mathcal{G}_{(1)}(\mathcal{C}^T\mathcal{C}\otimes\mathcal{C}^T\mathcal{C})\mathcal{G}_{(1)}^TVS$ is still a diagonal matrix. We just need to establish the fact that three of the diagonal entries are the same.

For one camera, $\mathcal{P}^T\mathcal{P}$ equals

$$\begin{bmatrix} 1 & 0 & P_i^2\cdot P_i^4 & 0 & -P_i^1\cdot P_i^4 & 0 \\ & 1 & P_i^3\cdot P_i^4 & 0 & 0 & -(P_i^1\cdot P_i^4) \\ & & P_i^4\cdot P_i^4-(P_i^1\cdot P_i^4)(P_i^1\cdot P_i^4) & 0 & -(P_i^1\cdot P_i^4)(P_i^4\cdot P_i^2) & -(P_i^1\cdot P_i^4)(P_i^4\cdot P_i^3) \\ & & & 1 & P_i^3\cdot P_i^4 & -P_i^2\cdot P_i^4 \\ & & & & (P_i^4\cdot P_i^4)-(P_i^2\cdot P_i^4)(P_i^4\cdot P_i^2) & -(P_i^2\cdot P_i^4)(P_i^4\cdot P_i^3) \\ & & & & & (P_i^4\cdot P_i^4)-(P_i^3\cdot P_i^4)(P_i^4\cdot P_i^3) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & P_i^2\cdot P_i^4 & 0 & -P_i^1\cdot P_i^4 & 0 \\ & 1 & P_i^3\cdot P_i^4 & 0 & 0 & -(P_i^1\cdot P_i^4) \\ & & \|P_i^4\|^2-\|P_i^1\cdot P_i^4\|^2 & 0 & -(P_i^1\cdot P_i^4)(P_i^4\cdot P_i^2) & -(P_i^1\cdot P_i^4)(P_i^4\cdot P_i^3) \\ & & & 1 & P_i^3\cdot P_i^4 & -P_i^2\cdot P_i^4 \\ & & & & \|P_i^4\|^2-\|P_i^2\cdot P_i^4\|^2 & -(P_i^2\cdot P_i^4)(P_i^4\cdot P_i^3) \\ & & & & & \|P_i^4\|^2-\|P_i^3\cdot P_i^4\|^2 \end{bmatrix}.$$

where the matrix is symmetric and we reduce redundancy by omitting the entries below the diagonal. For n cameras,

$$\mathcal{P}^T\mathcal{P} = \begin{bmatrix} n & 0 & \sum_i P_i^2\cdot P_i^4 & 0 & -\sum_i P_i^1\cdot P_i^4 & 0 \\ & n & \sum_i P_i^3\cdot P_i^4 & 0 & 0 & -\sum_i P_i^1\cdot P_i^4 \\ & & \sum_i\|P_i^4\|^2-\|P_i^1\cdot P_i^4\|^2 & 0 & -\sum_i(P_i^1\cdot P_i^4)(P_i^4\cdot P_i^2) & -\sum_i(P_i^1\cdot P_i^4)(P_i^4\cdot P_i^3) \\ & & & n & \sum_i P_i^3\cdot P_i^4 & -\sum_i P_i^2\cdot P_i^4 \\ & & & & \sum_i\|P_i^4\|^2-\|P_i^2\cdot P_i^4\|^2 & -\sum_i(P_i^2\cdot P_i^4)(P_i^4\cdot P_i^3) \\ & & & & & \sum_i\|P_i^4\|^2-\|P_i^3\cdot P_i^4\|^2 \end{bmatrix}.$$

where $P_i^a\cdot P_i^b$ means the dot product between the ath and bth column

The SVD of $\mathcal{P}^T\mathcal{P}$ is $\mathcal{P}^T\mathcal{P}=HDH^T$, where H is 6×6 orthonormal matrix, D is 6×6 diagonal matrix. However, since we have an $nI_{3\times 3}$ submatrix in $\mathcal{P}^T\mathcal{P}$, we deduce that n appears as an eigenvalue 3 times for $\mathcal{P}^T\mathcal{P}$, where we can use the determinant identity for block matrices. We check that this indeed holds by a computer calculation, generating random instances of P_i's and calculating the eigenvalues for $\mathcal{P}^T\mathcal{P}$.

As a result, in the thin SVD of $\mathcal{P}$, we have $\mathcal{P}=USV^T$ where $S=\sqrt{D}$, $V=H$. Then in

$$\begin{aligned} T_{(1)}^n(T_{(1)}^n)^T &= \mathcal{P}\mathcal{G}_{(1)}(\mathcal{C}^T\mathcal{C}\otimes\mathcal{C}^T\mathcal{C})\mathcal{G}_{(1)}^T\mathcal{P}^T \\ &= U(SV^T\mathcal{G}_{(1)}(\mathcal{C}^T\mathcal{C}\otimes\mathcal{C}^T\mathcal{C})\mathcal{G}_{(1)}^TVS)U^T, \end{aligned}$$

we see that $SV^T\mathcal{G}_{(1)}(\mathcal{C}^T\mathcal{C}\otimes\mathcal{C}^T\mathcal{C})\mathcal{G}_{(1)}^TVS$ is a diagonal matrix where three of the entries are the same. By the uniqueness of the eigenvalues, we see that we have a spectral decomposition of $T_{(1)}^n(T_{(1)}^n)^T$, so that three of the singular values of $T_{(1)}^n$ are equal. □

A.4 Proof details for Theorem 2

Proof. Note blockwise multiplication by a rank-1 tensor with nonzero entries preserves multilinear rank, since it is a Tucker product by invertible diagonal matrices. Therefore, without loss of generality we may assume $\lambda_{i11}=\lambda_{1j1}=\lambda_{11k}=1$ for all $i,j,k\in\{2,3,...,n\}$. Below we will prove it follows $\lambda_{ijk}=c$ if exactly one of i,j,k equals 1, and $\lambda_{ijk}=c^2$ if none of i,j,k equal 1 and the indices are not all the same, for some constant $c\in\mathbb{R}^*$. This will immediately imply the theorem, because taking $\alpha=\beta=(1cc...c)$ and $\gamma=(\frac{1}{c}11...1)$ achieves $\lambda_{ijk}=\alpha_i\beta_j\gamma_k$ whenever i,j,k are not all the same.

We consider the matrix flattenings $T_{(2)}^n$ and $(\lambda\odot_b T^n)_{(2)}$ in $\mathbb{R}^{3n\times 9n^2}$ of the block trifocal tensor and its scaled counterpart, with rows corresponding to the second mode of the tensors. By Theorem 1 and assumptions, the flattenings have matrix rank 4, thus all of their 5×5 minors vanish. The argument consists of considering several carefully chosen 5×5 submatrices of $(\lambda\odot_b T^n)_{(2)}$ to prove the existence of a constant c as above. Index the rows and columns of the flattenings by (j,r) and (iq,ks) respectively, for $i,j,k\in[n]$ and $q,r,s\in[3]$, so that e.g., $((\lambda\odot_b T^n)_{(2)})_{(j,r),(iq,ks)}=\lambda_{ijk}(T_{ijk}^n)_{qrs}$.

Step 1: The first submatrix of $(\lambda \odot_b T^n)_{(2)}$ we consider has column labels $(i1,11)$, $(i1,21)$, $(i1,31)$, $(i1,12)$, $(i2,11)$ and row labels $(1,1)$, $(1,2)$,$(1,3)$,$(i,1)$,$(i,2)$, where $i \in \{2,...,n\}$. Explicitly, it is

$$\begin{bmatrix} (T^n_{i11})_{111} & (T^n_{i11})_{211} & (T^n_{i11})_{311} & (T^n_{i11})_{112} & (T^n_{11i})_{111} \\ (T^n_{i11})_{121} & (T^n_{i11})_{221} & (T^n_{i11})_{321} & (T^n_{i11})_{122} & (T^n_{11i})_{121} \\ (T^n_{i11})_{131} & (T^n_{i11})_{231} & (T^n_{i11})_{331} & (T^n_{i11})_{132} & (T^n_{11i})_{131} \\ \lambda_{ii1}(T^n_{ii1})_{111} & \lambda_{ii1}(T^n_{ii1})_{211} & \lambda_{ii1}(T^n_{ii1})_{311} & \lambda_{ii1}(T^n_{ii1})_{112} & \lambda_{1ii}(T^n_{1ii})_{111} \\ \lambda_{ii1}(T^n_{ii1})_{121} & \lambda_{ii1}(T^n_{ii1})_{221} & \lambda_{ii1}(T^n_{ii1})_{321} & \lambda_{ii1}(T^n_{ii1})_{122} & \lambda_{1ii}(T^n_{1ii})_{121} \end{bmatrix},$$

which we abbreviate as

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ \lambda_{ii1}* & \lambda_{ii1}* & \lambda_{ii1}* & \lambda_{ii1}* & \lambda_{1ii}* \\ \lambda_{ii1}* & \lambda_{ii1}* & \lambda_{ii1}* & \lambda_{ii1}* & \lambda_{1ii}* \end{bmatrix}, \tag{12}$$

with asterisk denoting the corresponding entry in $T^n_{(2)}$. As a function of λ_{ii1},λ_{1ii}, the determinant of (12) is a degree ≤ 2 polynomial, which must be divisible by λ_{ii1} and $\lambda_{ii1}-\lambda_{1ii}$ (because if $\lambda_{ii1}=0$ then clearly the bottom two rows of (12) are linearly independent, and if $\lambda_{ii1}-\lambda_{1ii}=0$ we have a submatrix of $T^n_{(2)}$ with the bottom two rows scaled uniformly). So the determinant of (12) is a scalar multiple of $\lambda_{ii1}(\lambda_{ii1}-\lambda_{1ii})$. Note that the multiple is a polynomial function of the cameras P_1 and P_i. We claim that generically the multiple is nonzero; and to see this, it suffices to exhibit a *single* instance of (calibrated) cameras where the determinant of (12) does not vanish identically for all λ_{ii1},λ_{1ii} due to the polynomiality (e.g., see [53]). We check that this indeed holds by a computer calculation, generating numerical instances of P_1 and P_i randomly. Thus the vanishing of the minor in (12) implies $\lambda_{ii1}(\lambda_{ii1}-\lambda_{1ii})=0$, whence $\lambda_{ii1}=\lambda_{1ii}$ since $\lambda_{ii1}\neq 0$. An analogous calculation with $(\lambda \odot_b T^n)_{(3)}$ gives $\lambda_{i1i}=\lambda_{1ii}$.

Step 2: Next consider the submatrix of $(\lambda \odot_b T^n)_{(2)}$ with column labels $(j1,11)$, $(j1,21)$, $(j1,31)$, $(j1,12)$, $(1j,11)$ and row labels $(1,1)$, $(1,2)$, $(1,3)$, $(i,1)$, $(i,2)$, where $i,j \in \{2,...,n\}$ are distinct. It looks like

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{1ij}* \\ \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{1ij}* \end{bmatrix}, \tag{13}$$

with asterisks denoting entries of $T^n_{(2)}$. Similarly to the previous case, the determinant of (13) must be a scalar multiple of $\lambda_{ji1}(\lambda_{ji1}-\lambda_{1ij})$ where the scale depends polynomially on P_1,P_i,P_j. By a computer computation, we find that the scale is nonzero for random instances of cameras (alternatively, note the polynomial system in step 1 is a special case of the present one). It the scale is generically nonzero, hence $\lambda_{ji1}=\lambda_{1ij}$. An analogous calculation with $(\lambda \odot_b T^n)_{(3)}$ gives $\lambda_{i1j}=\lambda_{1ij}$.

Step 3: Consider the submatrix of $(\lambda \odot_b T^n)_{(2)}$ with column labels $(j1,11)$, $(j1,21)$, $(j1,31)$, $(j1,12)$, $(1k,11)$ and row labels $(1,1)$, $(1,2)$, $(1,3)$, $(i,1)$, $(i,2)$, for $i,j,k \in \{2,...,n\}$ distinct. It looks like

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{1ik}* \\ \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{1ik}* \end{bmatrix}. \tag{14}$$

The determinant of (14) is a scalar multiple of $\lambda_{ji1}(\lambda_{ji1}-\lambda_{1ik})$. By a direct computer computation as before, it is a nonzero multiple generically (alternatively, note the polynomial system in step 1 is a special of the present one). We deduce $\lambda_{ji1}=\lambda_{1ik}$. An analogous calculation with $(\lambda \odot_b T^n)_{(3)}$ gives $\lambda_{i1j}=\lambda_{1kj}$.

In particular, combining with step 2 it follows $\lambda_{1ij}=\lambda_{1ji}$, because $\lambda_{1ij}=\lambda_{k1j}=\lambda_{1kj}=\lambda_{ik1}=\lambda_{1ki}=\lambda_{j1i}=\lambda_{1ji}$. From this, step 1 and step 2, we have that the λ-scale does not depend on the ordering of its indices, provided there is a 1 among the indices.

Step 4: Consider the submatrix of $(\lambda \odot_b T^n)_{(2)}$ with column labels $(j1,11)$, $(j1,21)$, $(j1,31)$, $(j1,12)$, $(1i,11)$ and row labels $(1,1)$, $(1,2)$, $(1,3)$, $(i,1)$, $(i,2)$, for $i,j \in \{2,...,n\}$ distinct. It looks like

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{1ii}* \\ \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{ji1}* & \lambda_{1ii}* \end{bmatrix}. \tag{15}$$

The determinant of (15) is a scalar multiple of $\lambda_{ji1}(\lambda_{ji1} - \lambda_{1ii})$. By a direct computer computation, it is a nonzero multiple generically (alternatively, note the polynomial system in step 1 is a special case of the present one). We deduce $\lambda_{ji1} = \lambda_{1ii}$.

Putting together what we know so far, all λ-scales with a single 1-index agree. Indeed, this follows from $\lambda_{1ii} = \lambda_{ji1} = \lambda_{ij1} = \lambda_{1jj}$ so all λ-scales with a single 1-index and two repeated indices agree, combined with $\lambda_{ji1} = \lambda_{1ii}$ and the last sentence of step 3. Let $c \in \mathbb{R}^*$ denote this common scale.

Step 5: Consider the submatrix of $(\lambda \odot_b T^n)_{(2)}$ with column labels $(1i,11)$, $(1i,21)$, $(1i,31)$, $(1i,12)$, $(ij,11)$ and row labels $(1,1)$, $(1,2)$, $(1,3)$, $(i,1)$, $(i,2)$, for $i,j \in \{2,...,n\}$ distinct. It looks like

$$\begin{bmatrix} * & * & * & * & c* \\ * & * & * & * & c* \\ * & * & * & * & c* \\ c* & c* & c* & c* & \lambda_{iij}* \\ c* & c* & c* & c* & \lambda_{iij}* \end{bmatrix}. \tag{16}$$

As a function of c and λ_{iij}, the determinant of (16) is a scalar multiple of $c(c^2 - \lambda_{iij})$ (the second factor is present because it corresponds to scaling the bottom two rows and rightmost column of a 5×5 submatrix of $T_{(2)}$ each by c, which preserves rank deficiency). By a direct computer computation, we find that the scale is nonzero for a random instance of P_1, P_i, P_j, therefore it is nonzero generically. It follows $c^2 = \lambda_{iij}$. An analogous calculation with $(\lambda \odot_b T^n)_{(3)}$ gives $c^2 = \lambda_{iji}$.

Step 6: Consider the submatrix of $(\lambda \odot_b T^n)_{(2)}$ with column labels $(1i,11)$, $(1i,21)$, $(1i,31)$, $(1i,12)$, $(ji,11)$ and row labels $(1,1)$, $(1,2)$, $(1,3)$, $(i,1)$, $(i,2)$, for $i,j \in \{2,...,n\}$ distinct. It looks like

$$\begin{bmatrix} * & * & * & * & c* \\ * & * & * & * & c* \\ * & * & * & * & c* \\ c* & c* & c* & c* & \lambda_{jii}* \\ c* & c* & c* & c* & \lambda_{jii}* \end{bmatrix}. \tag{17}$$

Similarly to the previous step, the determinant of (17) must be a scalar multiple of $c(c^2 - \lambda_{jii})$. By a direct computer computation, it is a nonzero multiple generically. We deduce $c^2 = \lambda_{jii}$.

Step 7: Consider the submatrix of $(\lambda \odot_b T^n)_{(2)}$ with column labels $(1i,11)$, $(1i,21)$, $(1i,31)$, $(1i,12)$, $(ik,11)$ and row labels $(1,1)$, $(1,2)$, $(1,3)$, $(j,1)$, $(j,2)$, for $i,j,k \in \{2,...,n\}$ distinct. It looks like

$$\begin{bmatrix} * & * & * & * & c* \\ * & * & * & * & c* \\ * & * & * & * & c* \\ c* & c* & c* & c* & \lambda_{ijk}* \\ c* & c* & c* & c* & \lambda_{ijk}* \end{bmatrix}. \tag{18}$$

The determinant of (18) is a scalar multiple of $c(c^2 - \lambda_{ijk})$. By a direct computer computation, it is a nonzero multiple generically (alternatively, note the polynomial system in step 5 is a special case of the present case). We deduce $c^2 = \lambda_{ijk}$.

At this point, by steps 5,6,7 we have that all λ-scales with no 1-indices and not all indices the same must equal c^2. Combined with the second paragraph of step 4, this shows c satisfies the property announced at the start of the proof. Therefore the proof is complete. □

A.5 Implementation details

A.5.1 Estimating trifocal tensors from three fundamental matrices

Given three cameras P_1, P_2, P_3 and the corresponding fundamental matrices F_{21}, F_{31}, F_{32}, we can calculate the trifocal tensor T_{ijk} using the following procedure detailed in [1]. Specifically, from

F_{21} calculate an initial estimate of the cameras P_1', P_2'. Then, $P_3^T F_{32} P_2'$ and $P_3^T F_{31} P_1'$ should be skew-symmetric matrices. This gives 20 linear equations in terms of the entries in P_3, which can be used to solve for the trifocal tensor. Note that there are no geometrical constraints when calculating P_3, and there will be no guarantee of the quality of the estimation.

A.5.2 Higher-order regularized subspace-constrained Tyler's estimator (HOrSTE) for EPFL

We describe the robust variant of SVD that we used for the EPFL experiments in Section 4. Numerically, it performs more stably and accurately than HOSVD-HT, yet it is an iterative procedure and each iteration requires an SVD of the $3n \times 9n^2$ flattening. This becomes computationally expensive when n becomes large and the number of iterations are also large. However, since the number of cameras for the EPFL dataset are below 20 cameras, the computational overhead is not too great.

In HOSVD, a low dimensional subspace is estimated using the singular value decomposition and taking the R_n leading left singular vectors for each mode-n flattening. The Tyler's M Estimator (TME) [40] is a robust covariance estimator of a D dimensional dataset $\{x_i\}_{i=1}^N \subset \mathbb{R}^D$. It minimizes the objective function

$$\min_{\Sigma \in \mathbb{R}^{D \times D}} \frac{D}{N} \sum_{i=1}^{N} \log(x_i^T \Sigma^{-1} x_i) + \log\det(\Sigma) \tag{19}$$

such that Σ is positive definite and has trace 1. The TME estimator can be applied to robustly find an R_n dimensional subspace by taking the R_n leading eigenvectors of the covariance matrix of TME. To compute the TME, [40] proposes an iterative algorithm, where

$$\Sigma^{(k)} = \sum_{i=1}^{N} \frac{x_i x_i^T}{x_i^T (\Sigma^{(k-1)})^{-1}) x_i} / tr(\sum_{i=1}^{N} \frac{x_i x_i^T}{x_i^T (\Sigma^{(k-1)})^{-1}) x_i}). \tag{20}$$

TME doesn't exist when $D > N$, but a regularized TME has been proposed by [54]. The iterations become

$$\Sigma^{(k)} = \frac{1}{1+\alpha} \frac{D}{N} \sum_{i=1}^{N} \frac{x_i x_i^T}{x_i^T (\Sigma^{(k-1)})^{-1}) x_i} + \frac{\alpha}{1+\alpha} I \tag{21}$$

where α is a regularization parameter, and I is the $D \times D$ identity matrix. TME does not assume the dimension of the subspace d is predetermined. In the case when d is prespecified, [41] improves the TME estimator by incorporating the information into the algorithm and develops the subspace-constrained Tyler's Estimator (STE). For each iteration, STE equalizes the trailing $D-d$ eigenvalues of the estimated covariance matrix and uses a parameter $0 < \gamma < 1$ to shrink the eigenvalues. The iterative procedure for STE is summarized into 3 steps:

1. Calculate the unnormalized TME, $Z^{(k)} = \sum_{i=1}^{N} (x_i x_i^T / x_i^T (\Sigma^{(k-1)})^{-1}) x_i)$.
2. Perform the eigendecomposition of $Z^{(k)} = U^{(k)} S^{(k)} (U^{(k)})^T$, and set the trailing $D-d$ eigenvalues as $\gamma \sum_{i=d+1}^{D} \sigma_i / (D-d)$.
3. Calculate $\Sigma^{(k)} = U^{(k)} S^{(k)} (U^{(k)})^T / tr(U^{(k)} S^{(k)} (U^{(k)})^T)$, which is the normalized covariance matrix. Repeat steps 1-3 until convergence.

Similar to the regularized TME, STE can also be regularized to succeed in situations where there are fewer inliers, and can improve the robustness of the algorithm. The *regularized STE* differs from STE in only the first step, which is replaced by

1.* Calculate the unnormalized regularized TME, $Z^{(k)} = \frac{1}{1+\alpha} \frac{D}{N} \sum_{i=1}^{N} \frac{x_i x_i^T}{x_i^T (\Sigma^{(k-1)})^{-1}) x_i} + \frac{\alpha}{1+\alpha} I$

We apply the regularized STE to the HOSVD framework, and call the resulting projection the *higher-order regularized STE* (HOrSTE). It is performed via the following steps:

1. For each n, calculate the factor matrices A_n as the R_n leading left singular vectors from regularized STE applied to $T_{(n)}$.
2. Set the core tensor $\mathcal{G}$ as $\mathcal{G} = T \times_1 A_1^T \times_2 A_2^T \cdots \times_N A_N^T$.

A.6 Additional numerical results

In this section, we include comprehensive results for the rotation and translation errors for the EPFL and Photo Tourism experiments. Table 1 and 2 contains all results for EPFL datasets. Table 3 contains the location estimation errors for Photo Tourism. Table 4 contains the rotation estimation errors for Photo Tourism. In Table 4, we only report the rotation errors for LUD for all the methods that we compared against, as they are mostly the same since they used the same rotation averaging method.

Table 1: EPFL synchronization errors. $\bar{e}_r$ is the mean rotation error in degrees, $\hat{e}_r$ is the median rotation error in degrees. $\bar{e}_t$ is the mean location error, $\hat{e}_t$ is the median location error. NRFM(LUD) is NRFM initialized with LUD and NRFM is randomly initialized. BATA(MLPS) is BATA initialized with MPLS.

	Our		LUD		NRFM(LUD)		NRFM		BATA(MPLS)	
Dataset	$\bar{e}_t$	$\hat{e}_t$	$\bar{e}_t$	$\hat{e}_t$	$\bar{e}_t$	$\hat{e}_t$	$\bar{e}_t$	$\hat{e}_t$	$\bar{e}_t$	$\hat{e}_t$
FountainP11	**0.008**	**0.007**	0.91	0.54	0.75	0.46	3.37	3.03	1.12	1.01
HerzP8	**0.02**	**0.02**	5.06	5.06	4.37	3.42	4.24	3.14	5.04	5.03
HerzP25	**4.70**	**4.68**	7.75	8.00	6.20	5.82	8.85	8.38	7.77	8.41
EntryP10	**0.05**	**0.02**	3.08	3.02	1.34	1.11	7.63	7.43	2.90	2.58
CastleP19	9.64	5.80	4.58	4.04	**3.37**	**3.02**	15.81	15.43	5.77	5.62
CastleP30	11.00	11.33	4.27	3.72	**3.24**	**2.75**	16.54	17.04	4.23	3.26

Table 2: EPFL synchronization errors. $\bar{e}_r$ is the mean rotation error in degrees, $\hat{e}_r$ is the median rotation error in degrees. BATA(MLPS) is BATA initialized with MPLS.

	Our		LUD		BATA(MPLS)	
Dataset	$\bar{e}_r$	$\hat{e}_r$	$\bar{e}_r$	$\hat{e}_r$	$\bar{e}_r$	$\hat{e}_r$
FountainP11	0.09	0.08	**0.05**	**0.05**	0.06	0.05
HerzP8	**0.12**	**0.12**	0.33	0.34	0.44	0.39
HerzP25	2.01	1.11	**0.18**	**0.19**	0.26	0.23
EntryP10	**0.15**	**0.11**	0.25	0.25	0.27	0.25
CastleP19	56.24	11.71	**0.24**	**0.22**	0.27	0.25
CastleP30	38.84	4.58	**0.13**	**0.13**	0.19	0.15

Table 3: Translation errors for Photo Tourism. n is the size after downsampling. Est. % is the ratio of observed blocks over total number of blocks. $\bar{e}_t$ is the mean location error, $\hat{e}_t$ is the median location error. NRFM(L) is NRFM initialized with LUD and NRFM(R) is randomly initialized. The notation PR means that the dataset was further downsampled to match the two view methods. BATA is BATA initialized with MPLS. We were not able to get results for our subsampled dataset for Piccadilly with MPLS.

Dataset	Our Approach				NRFM(L)		LUD		NRFM(R)		BATA	
dataset	n	Est. %	$\bar{e}_t$	$\hat{e}_t$	$\bar{e}_t$	$\hat{e}_t$	$\bar{e}_t$	$\hat{e}_t$	$\bar{e}_t$	$\hat{e}_t$	$\bar{e}_t$	$\hat{e}_t$
Piazza del Popolo	185	72.3	**0.78**	**0.45**	1.63	0.85	1.66	0.86	13.45	12.06	1.63	1.10
NYC Library	127	64.7	**1.01**	**0.53**	1.39	0.48	1.49	0.57	13.06	14.03	1.59	0.68
Ellis Island	194	70.3	**9.56**	**7.73**	19.31	16.97	20.71	17.96	26.08	26.38	23.63	22.50
Tower of London	130	34.1	4.15	2.66	3.26	2.49	3.54	2.51	49.99	47.33	**2.70**	**2.26**
Madrid Metropolis	190	35.9	18.93	15.53	**1.91**	**1.19**	1.94	1.20	31.48	24.02	3.33	1.72
Yorkminster	196	37.2	1.46	**1.14**	2.31	1.39	2.35	1.45	16.67	14.46	**1.37**	1.15
Alamo	224	94.3	0.62	**0.28**	**0.53**	0.31	**0.53**	0.31	10.04	7.68	0.55	0.33
Vienna Cathedral	197	97.8	**0.73**	**0.33**	2.96	1.64	3.15	1.79	16.08	14.76	6.16	2.18
Roman Forum(PR)	111	51.1	10.71	6.75	**1.59**	**0.89**	1.63	0.93	23.23	11.20	1.85	1.04
Notre Dame	214	96.6	0.57	0.34	**0.38**	**0.21**	0.38	0.21	6.87	4.75	1.02	0.26
Montreal N.D.	162	97.0	**0.38**	**0.24**	0.56	0.37	0.57	0.38	10.33	11.15	0.58	0.41
Union Square	144	28.6	5.64	3.99	**4.31**	**3.76**	4.85	4.38	9.59	6.69	5.77	4.83
Gendarmenmarkt	112	89.7	45.34	23.63	37.93	17.35	**37.92**	17.41	62.69	26.42	54.38	**15.91**
Piccadilly(PR)	169	55.4	**0.73**	**0.39**	3.68	1.90	3.71	1.93	13.55	13.34	-	-

Table 4: Rotation errors for Photo Tourism. N is the total number of cameras. n is the size after down sampling. Est. % is the ratio of observed blocks over total number of blocks. $\bar{e}_r$ is the mean rotation error, $\hat{e}_r$ is the median rotation error. The notation PR means that the dataset was further down sampled to match the two view methods. We were not able to get results for our subsampled dataset for Piccadilly with MPLS.

	Our Approach					LUD		MPLS		
dataset	N	n	Est. %	$\bar{e}_r$	$\hat{e}_r$	$\bar{e}_r$	$\hat{e}_r$	$\bar{e}_r$	$\hat{e}_r$	Our Runtime (s)
Piazza del Popolo	307	185	72.3	1.26	0.61	0.72	0.43	**0.69**	**0.41**	13531
NYC Library	306	127	64.7	2.80	1.58	**1.16**	0.61	1.19	**0.57**	4465
Ellis Island	223	194	70.3	4.61	1.11	1.16	0.50	**0.99**	**0.49**	13816
Tower of London	440	130	34.1	2.28	1.31	**1.63**	**1.28**	1.66	1.37	4242
Madrid Metropolis	315	190	35.9	28.85	4.60	**1.27**	**0.61**	1.54	1.15	11764
Yorkminster	410	196	37.2	2.33	1.97	**1.34**	1.09	1.89	**1.04**	13115
Alamo	564	224	94.3	1.10	0.76	**1.07**	**0.68**	1.09	**0.68**	17513
Vienna Cathedral	770	197	97.8	0.74	0.46	0.40	**0.28**	**0.39**	**0.28**	12499
Roman Forum(PR)	989	111	51.1	11.86	3.39	**0.40**	**0.28**	1.07	0.65	2162
Notre Dame	547	214	96.6	0.78	0.50	**0.67**	**0.43**	0.68	**0.43**	17430
Montreal N.D.	442	162	97.0	0.50	0.35	**0.49**	0.32	**0.49**	**0.31**	7241
Union Square	680	144	28.6	20.70	5.29	**1.82**	**1.34**	2.00	1.56	4355
Gendarmenmarkt	655	112	89.7	22.95	15.24	18.42	10.25	**17.42**	**8.41**	2432
Piccadilly(PR)	1000	169	55.4	**2.01**	**0.96**	6.12	2.95	-	-	11230

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please check Section 3 and 4 for the main claims of our paper and the experimental results. An accurate summary of the contributions is found at the end of the introduction section, and also conveyed in the abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please check our conclusion in Section 5. The last paragraph describes the limitations of the work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings,

model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Please see the proofs in Appendix A.3, A.2, A.4 and statements in Section 2. The statements are precise, and accompanied by complete proofs.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we mention the numerical details and processing of our algorithm in section Section 4 for each experiment. We also include code in the supplementary material with instructions on how to run them. Our code depends on other publicly available code. Instructions can be found in supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Please feel free to check our code in the supplementary material and the README.MD file for instructions. A public version will be available in the future.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Please see our experiment details in Section 4. In particular, we included the way we chose hyperparameter and specific numbers when applicable.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Though we didn't set a random seed for some experiments, there were insignificant differences between different runs of the method.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Please see our experiment details in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We only used open source code and have complied to the code of ethics. Please check code in supplementary material.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our method is focused on a general framework for tensor based structure from motion and we run experiments on open source datasets. No negative social impacts are related to our work. Having a tensor based synchronization method opens up more research directions and could help apply structure from motion to develop applications where point correspondences are scarce.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper doesn't pose such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited the authors whose code in our code, and respected all relevant licenses. We include links to code in our submitted code as well. Please refer to the code and the experiment details.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide code for our methods and instructions for running the methods in the supplementary material.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper doesn't involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.